Engineering
Simplicity

IPsec VPN User Guide

Published
2023-09-28

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

IPsec VPN User Guide

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

3

PKI in Junos OS

6

Policy Based VPN

7

Route Based VPN

13 **Remote Access VPN**

14 **Quantum Safe VPN**

15 **Monitoring VPN**

About This Guide

Use this guide to configure, monitor, and manage the IPsec VPN feature on Junos OS devices to enable secure communications across a public WAN such as the Internet.

RELATED DOCUMENTATION

Learn About Secure VPNs

Configuring a Small Office for High-Definition Videoconferencing

Configuring Branch SRX Series for MPLS over GRE with IPsec Segmentation

1

CHAPTER

PKI Fundamentals

Public Key Infrastructure (PKI)

IN THIS SECTION

This topic describes the overview of public key infrastructure and includes the following sections:

Introduction to PKI

Public key infrastructure (PKI) provides a way of verifying the identity of a remote site by using a digital certificate. PKI uses a certificate authority (CA) to validate your information and to sign it with a digital signature such that neither your information nor the signature can be modified. Once signed, the information becomes a digital certificate. Devices that receive a digital certificate can verify the information in the certificate by validating the signature using public key cryptography.

The Public Key Infrastructure (PKI) provides an infrastructure for digital certificate management and consists of:

- Registration Authority (RA) that verifies the identities of entities, authorizes their certificate requests, and generates unique asymmetric key pairs (unless the users’ certificate requests already contain public keys)
- Certificate Authority (CA) that issues corresponding digital certificates for the requesting entities.
- A certificate revocation list (CRL) identifying the certificates that are no longer valid. Each entity possessing the authentic public key of a CA can verify the certificates issued by that CA.

Digital Certificate

A digital certificate is an electronic file that verifies the identity of the certificate's holder to protect data exchanged online. Digital certificates provide a way of authenticating users through a trusted third party called a certificate authority (CA). The CA validates the identity of a certificate holder and "signs" the certificate to attest that it has not been forged or altered. Alternatively, you can use a self-signed certificate to attest to your identity.

A key pair is a critical element of a digital certificate implementation. The public key is included in the local digital certificate and the private key is used to decrypt data received from peers.

Certificates have a finite lifetime and are defined by a start time and an end time. The certificate becomes invalid when the life time expires. When the certificate expires, a certificate renewal or a new certificate request is required.

Certificate Authority

A CA is a trusted third-party organization that creates, enrolls, validates, and revokes digital certificates. The CA guarantees a user's identity and issues public and private keys for message encryption and decryption (coding and decoding). A CA also generates certificate revocation lists (CRLs) which are lists of revoked certificates.

Private/Public Key Pair

When setting up a PKI, you must include Public and private keys that are generated in pairs and linked mathematically.

When request for the certificate, you must include the public key in the certificate enrollment request. The public key will be included in the granted certificate and the private key is kept on the requesting device. A message encrypted with the public key can be decrypted by using the corresponding private key. The private-public key pair is also used for creating digital signatures.

Certificate Enrollment Options

You can request a CA digital certificate either online or manually:

- Manual certificate enrollment—This process includes generation of a PKCS10 request, submission to the certificate authority (CA), retrieval of the signed certificate, and manually loading of the certificate into the Junos OS device as the local certificate.
- Online certificate enrollment—You can use either Certificate Management Protocol version 2 (CMPv2) or Simple Certificate Enrollment Protocol (SCEP) for online certificate enrollment.

Certificate Revocation Options

- Certificate revocation list (or CRL)—Certificate authority (CA) periodically publishes a list of revoked certificate using a certificate revocation list (CRL). The CRL contains the list of digital certificates with serial numbers that have been canceled before their expiration date.
- Online Certificate Status Protocol (OCSP)—OCSP is used to check the revocation status of X509 certificates. The OCSP provides revocation status on certificates in real time and is useful in time-sensitive situations such as bank transactions and stock trades

Certificate Request Types

Public Key Infrastructure (PKI) allows users to authenticate each other using digital certificates issued by CA. PKI Uses X.509, Public Key Cryptography Standards (PKCS) to define the standard formats for certificates and their use. In PKI, an applicant uses a certificate signing request (CSR) to apply for a digital certificate to a certificate authority (CA). The request can be in one of the standard:

- Public-Key Cryptography Standard # (PKCS#) (PKCS7, PKCS10, PKCS11, PKCS12)
- x509-signaturere.

Certificate Signatures and Verification

A digital certificate is an electronic means for verifying your identity through a trusted third party, known as a certificate authority (CA). Alternatively, you can use a self-signed certificate to attest to your identity.

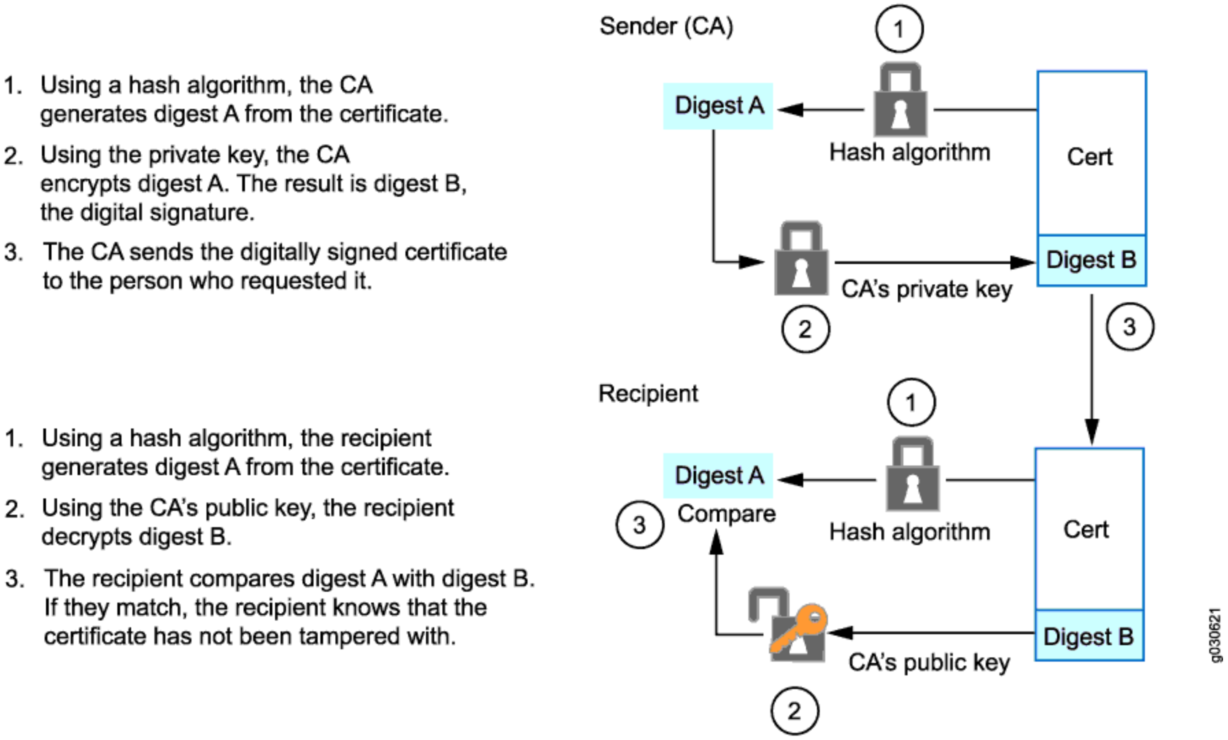
The CA server you use can be owned and operated by an independent CA or by your own organization, in which case you become your own CA. If you use an independent CA, you must contact them for the addresses of their CA and certificate revocation list (CRL) servers (for obtaining certificates and CRLs) and for the information they require when submitting personal certificate requests. When you are your own CA, you determine this information yourself.

The CA that issues a certificate uses a hash algorithm to generate a digest, and then “signs” the certificate by encrypting the digest with its private key. The result is a digital signature. The CA then makes the digitally signed certificate available for download to the person who requested it. Figure 1 on page 5 illustrates this process.

The recipient of the certificate generates another digest by applying the same hash algorithm to the certificate file, then uses the CA's public key to decrypt the digital signature. By comparing the decrypted digest with the digest just generated, the recipient can confirm the integrity of the CA's signature and, by extension, the integrity of the accompanying certificate. Figure 1 on page 5 illustrates this process.

A certificate is considered valid if the digital signature can be verified and the serial number of the certificate is not listed in a certificate revocation list.

Figure 1: Digital Signature Verification



1. Using a hash algorithm, the CA generates digest A from the certificate.
2. Using the private key, the CA encrypts digest A. The result is digest B, the digital signature.
3. The CA sends the digitally signed certificate to the person who requested it.

1. Using a hash algorithm, the recipient generates digest A from the certificate.
2. Using the CA’s public key, the recipient decrypts digest B.
3. The recipient compares digest A with digest B. If they match, the recipient knows that the certificate has not been tampered with.

When Digital Signature Algorithm (DSA) signatures are used, the SHA-1 hash algorithm is used to generate the digest. When Rivest-Shamir-Adleman (RSA) signatures are used, SHA-1 is the default hash algorithm used to generate the digest; you can specify the SHA-256 hash algorithm with the `digest` option of the `request security pki generate-certificate-request` or `request security pki local-certificate generate-self-signed` commands. When Elliptic Curve Digital Signature Algorithm (ECDSA) signatures are used, the SHA-256 hash algorithm is used for ECDSA-256 signatures and the SHA-384 hash algorithm is used for ECDSA-384 signatures.

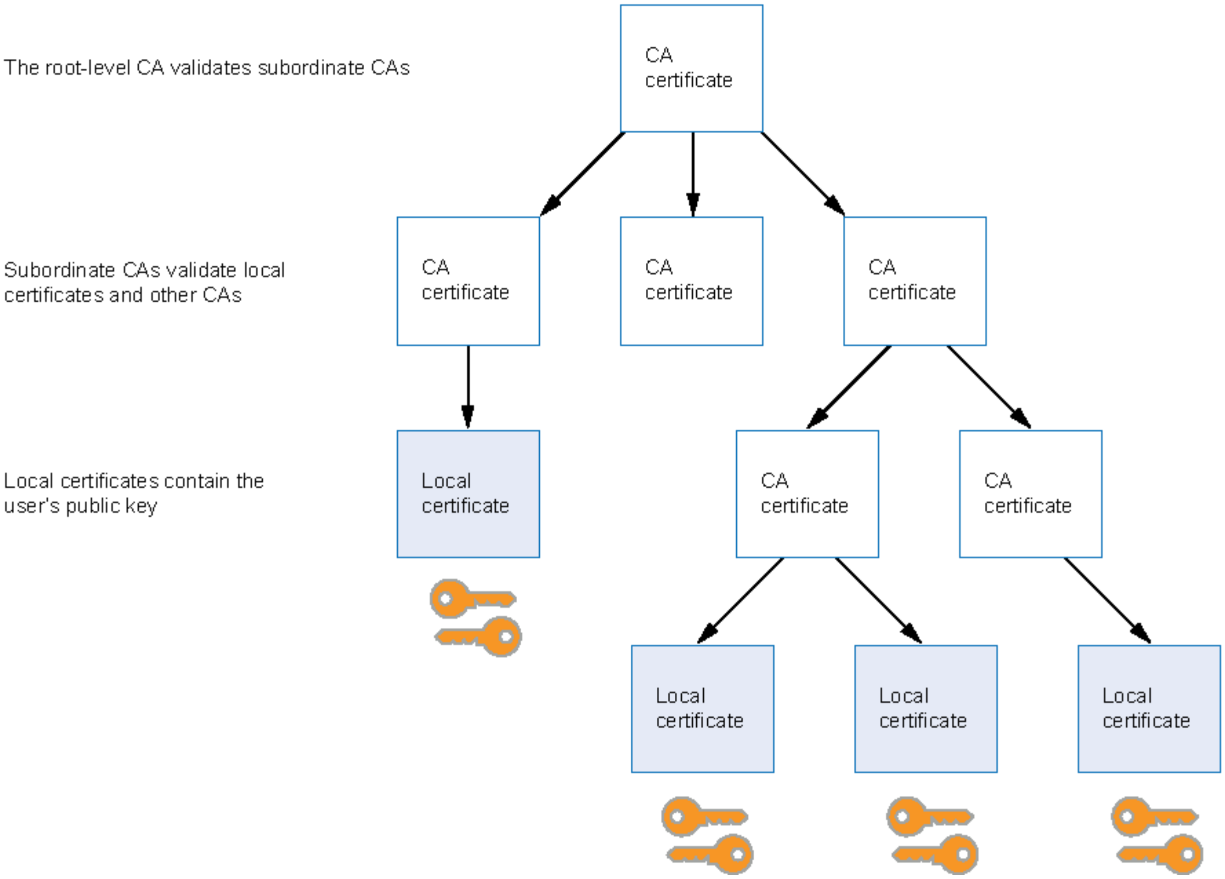
Starting in Junos OS Release 18.1R3, the default hash algorithm that is used for validating automatically and manually generated self-signed PKI certificates is Secure Hash Algorithm 256 (SHA-256). Prior to Junos OS Release 18.1R3, SHA-1 is used as default hash algorithm.

Certificate Validation

To verify the trustworthiness of a certificate, you must be able to track a path of certified certificate authorities (CAs) from the one issuing your local certificate to the root authority of a CA domain. Public key infrastructure (PKI) refers to the hierarchical structure of trust required for the successful implementation of public key cryptography.

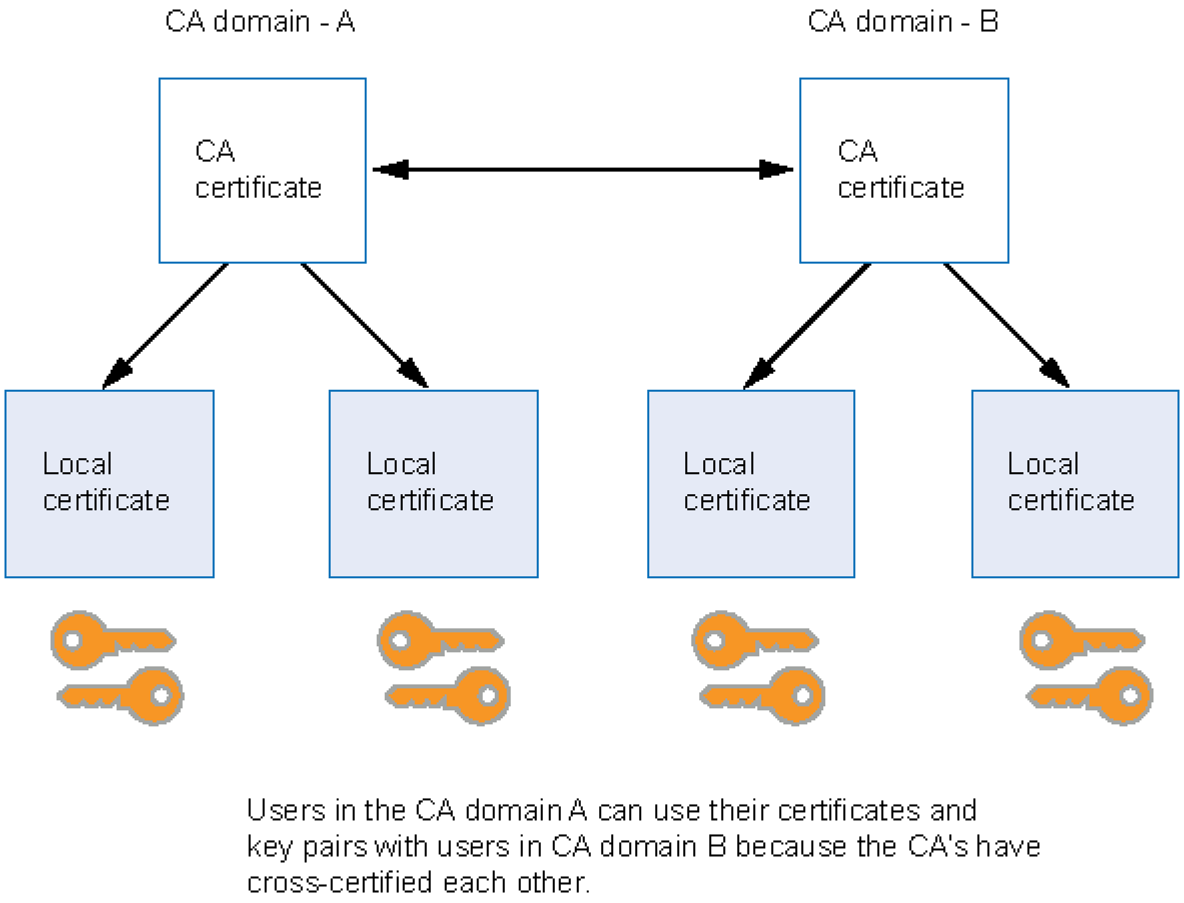
Figure 2 on page 7 shows the structure of a single-domain certificate authority with multiple hierarchy levels.

Figure 2: PKI Hierarchy of Trust—CA Domain



If certificates are used solely within an organization, that organization can have its own CA domain within which a company CA issues and validates certificates for its employees. If that organization later wants its employees to exchange their certificates with certificates from another CA domain (for example, with employees at another organization that has its own CA domain), the two CAs can develop cross-certification by agreeing to trust the authority of each other. In this case, the PKI structure does not extend vertically but does extend horizontally. See Figure 3 on page 8.

Figure 3: Cross-Certification


CA domain - A
CA domain - B
CA certificate
CA certificate
Local certificate
Local certificate
Local certificate
Local certificate
Users in the CA domain A can use their certificates and key pairs with users in CA domain B because the CA's have cross-certified each other.

Release History Table

Release	Description
18.1R3	Starting in Junos OS Release 18.1R3, the default hash algorithm that is used for validating automatically and manually generated self-signed PKI certificates is Secure Hash Algorithm 256 (SHA-256). Prior to Junos OS Release 18.1R3, SHA-1 is used as default hash algorithm.

2

CHAPTER

IPsec Fundamentals

Internet Key Exchange

IN THIS SECTION

Introduction to IKE

Internet Key Exchange (IKE) is a secure key management protocol that is used to set up a secure, authenticated communications channel between two devices.

IKE does the following:

- Negotiates and manages IKE and IPsec parameters
- Authenticates secure key exchange
- Provides mutual peer authentication by means of shared secrets (not passwords) and public keys
- Provides identity protection (in main mode)
- Employs Diffie-Hellman methods and is optional in IPsec (the shared keys can be entered manually at the endpoints).

IKE Versions

Two versions of the IKE standards are available:

- IKE version 1 - IKE protocol defined in RFC 2409.
- IKE version 2 - IKE version 2 (IKEv2) is the latest version of the IKE protocol defined in RFC 7296.

Internet Key Exchange version 2 (IKEv2) is the latest version of the Internet Key Exchange (IKE) protocol defined in RFC 7296. A VPN peer is configured as either IKEv1 or IKEv2. When a peer is configured as IKEv2, it cannot fall back to IKEv1 if its remote peer initiates IKEv1 negotiation.

The advantages of using IKEv2 over IKEv1 are as follows:

- Replaces eight initial exchanges with a single four-message exchange.
- Reduces the latency for the IPsec SA setup and increases connection establishment speed.
- Increases robustness against DOS attacks.
- Improves reliability through the use of sequence numbers, acknowledgments, and error correction.
- Improves reliability, as all messages are requests or responses. The initiator is responsible for retransmitting if it does not receive a response.

Interaction Between IKE and IPSec

IPsec can establish a VPN in either of the following way:

- Internet Key Exchange (IKE) protocol— IPsec supports automated generation and negotiation of keys and security associations using the IKE protocol. Using IKE to negotiate VPNs between two endpoints provides more security than the manual key exchange.
- Manual key exchange—IPsec supports using and exchanging of keys manually (example: phone or email) on both sides to establish VPN.

IKEv1 Message Exchange

IKE negotiation includes two phases:

- Phase 1—Negotiate exchange of proposals for how to authenticate and secure the channel.

- Phase 2—Negotiate security associations (SAs) to secure the data that traverses through the IPsec tunnel.

Phase 1 of IKE Tunnel Negotiation

IN THIS SECTION

Phase 1 of an AutoKey Internet Key Exchange (IKE) tunnel negotiation consists of the exchange of proposals for how to authenticate and secure the channel. The participants exchange proposals for acceptable security services such as:

- Encryption algorithms—Data Encryption Standard (DES), triple Data Encryption Standard (3DES), and Advanced Encryption Standard (AES). (See "IPsec Overview" on page 20.)
- Authentication algorithms—Message Digest 5 (MD5) and Secure Hash Algorithm (SHA). (See "IPsec Overview" on page 20.)
- Diffie-Hellman (DH) group. (See "IPsec Overview" on page 20.)
- Preshared key or RSA/DSA certificates. (See "IPsec Overview" on page 20.)

A successful Phase 1 negotiation concludes when both ends of the tunnel agree to accept at least one set of the Phase 1 security parameters proposed and then process them. Juniper Networks devices support up to four proposals for Phase 1 negotiations, allowing you to define how restrictive a range of security parameters for key negotiation you will accept. Junos OS provides predefined standard, compatible, and basic Phase 1 proposal sets. You can also define custom Phase 1 proposals.

Phase 1 exchanges can take place in either main mode or aggressive mode. You can choose your mode during IKE policy configuration.

This topic includes the following sections:

Main Mode

In main mode, the initiator and recipient send three two-way exchanges (six messages total) to accomplish the following services:

- First exchange (messages 1 and 2)—Proposes and accepts the encryption and authentication algorithms.
- Second exchange (messages 3 and 4)—Executes a DH exchange, and the initiator and recipient each provide a pseudorandom number.
- Third exchange (messages 5 and 6)—Sends and verifies the identities of the initiator and recipient.

The information transmitted in the third exchange of messages is protected by the encryption algorithm established in the first two exchanges. Thus, the participants' identities are encrypted and therefore not transmitted "in the clear."

Aggressive Mode

In aggressive mode, the initiator and recipient accomplish the same objectives as with main mode, but in only two exchanges, with a total of three messages:

- First message—The initiator proposes the security association (SA), initiates a DH exchange, and sends a pseudorandom number and its IKE identity.

  When configuring aggressive mode with multiple proposals for Phase 1 negotiations, use the same DH group in all proposals because the DH group cannot be negotiated. Up to four proposals can be configured.

- Second message—The recipient accepts the SA; authenticates the initiator; and sends a pseudorandom number, its IKE identity, and, if using certificates, the recipient's certificate.
- Third message—The initiator authenticates the recipient, confirms the exchange, and, if using certificates, sends the initiator's certificate.

Because the participants' identities are exchanged in the clear (in the first two messages), aggressive mode does not provide identity protection.

Main and aggressive modes applies only to IKEv1 protocol. IKEv2 protocol does not negotiate using main and aggressive modes.

SEE ALSO

Understanding IKE Phase 1 Configuration for Group VPNv1 | **714**

proposal-set (Security IKE) | **1607**

Phase 2 of IKE Tunnel Negotiation

IN THIS SECTION

After the participants have established a secure and authenticated channel, they proceed through Phase 2, in which they negotiate security associations (SAs) to secure the data to be transmitted through the IPsec tunnel.

Similar to the process for Phase 1, the participants exchange proposals to determine which security parameters to employ in the SA. A Phase 2 proposal also includes a security protocol—either Encapsulating Security Payload (ESP) or Authentication Header (AH)—and selected encryption and authentication algorithms. The proposal can also specify a Diffie-Hellman (DH) group, if Perfect Forward Secrecy (PFS) is desired.

Regardless of the mode used in Phase 1, Phase 2 always operates in quick mode and involves the exchange of three messages.

This topic includes the following sections:

Proxy IDs

In Phase 2, the peers exchange proxy IDs. A proxy ID consists of a local and remote IP address prefix. The proxy ID for both peers must match, which means that the local IP address specified for one peer must be the same as the remote IP address specified for the other peer.

Perfect Forward Secrecy

PFS is a method for deriving Phase 2 keys independent from and unrelated to the preceding keys. Alternatively, the Phase 1 proposal creates the key (the SKEYID_d key) from which all Phase 2 keys are derived. The SKEYID_d key can generate Phase 2 keys with a minimum of CPU processing. Unfortunately, if an unauthorized party gains access to the SKEYID_d key, all your encryption keys are compromised.

PFS addresses this security risk by forcing a new DH key exchange to occur for each Phase 2 tunnel. Using PFS is thus more secure, although the rekeying procedure in Phase 2 might take slightly longer with PFS enabled.

Replay Protection

A replay attack occurs when an unauthorized person intercepts a series of packets and uses them later either to flood the system, causing a denial of service (DoS), or to gain entry to the trusted network. Junos OS provides a replay protection feature that enables devices to check every IPsec packet to see if it has been received previously. If packets arrive outside a specified sequence range, Junos OS rejects them. Use of this feature does not require negotiation, because packets are always sent with sequence numbers. You simply have the option of checking or not checking the sequence numbers.

SEE ALSO

Understanding IPsec SA Configuration for Group VPNv2 | **765**

policy (Security IPsec) | **1583**

IKEv2 Message Exchange

IN THIS SECTION

IKE version 2 is the successor to the IKEv1 method. It provides a secure VPN communication channel between peer VPN devices and defines negotiation and authentication for IPsec security associations (SAs) in a protected manner.

IKEv2 does not include phase 1 and phase 2 similar to IKEv1, but there are four message exchanges occur to negotiate an IPsec tunnel with IKEv2. The message exchange in IKEv2 are:

- Negotiates the security attributes to establish the IPsec tunnel. This includes exchanging the protocols/parameters used, and Diffie-Hellman groups.

- Each peer establishes or authenticates their identities while the IPsec tunnel is established.
- Peers to create additional security associations between each other.
- Peers perform liveliness detection, removing SA relationships, and reporting error messages.

IKEv2 Configuration Payload

Configuration payload is an IKEv2 option offered to propagate provisioning information from a responder to an initiator. IKEv2 configuration payload is supported with route-based VPNs only.

RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*, defines 15 different configuration attributes that can be returned to the initiator by the responder.

IKEv2 Rekeying and Reauthentication

With IKEv2, rekeying and reauthentication are separate processes.

Rekeying establishes new keys for the IKE security association (SA) and resets message ID counters, but it does not reauthenticate the peers.

Reauthentication verifies that VPN peers retain their access to authentication credentials. Reauthentication establishes new keys for the IKE SA and child SAs; rekeys of any pending IKE SA or child SA are no longer needed. After the new IKE and child SAs are created, the old IKE and child SAs are deleted.

IKEv2 reauthentication is disabled by default. You enable reauthentication by configuring a reauthentication frequency value between 1 and 100. The reauthentication frequency is the number of IKE rekeys that occurs before reauthentication occurs. For example, if the configured reauthentication frequency is 1, reauthentication occurs every time there is an IKE rekey. If the configured reauthentication frequency is 2, reauthentication occurs at every other IKE rekey. If the configured reauthentication frequency is 3, reauthentication occurs at every third IKE rekey, and so on.

IKEv2 Fragmentation

When certificate-based authentication is used, IKEv2 packets can exceed the path MTU if multiple certificates are transmitted. If the IKE message size exceeds the path MTU, the messages are fragmented at the IP level. Some network equipment, such as NAT devices, does not allow IP fragments to pass through, which prevents the establishment of IPsec tunnels.

IKEv2 message fragmentation, as described in RFC 7383, Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation, allows IKEv2 to operate in environments where IP fragments might be blocked and peers would not be able to establish an IPsec security association (SA). IKEv2 fragmentation splits a large IKEv2 message into a set of smaller ones so that there is no fragmentation at the IP level. Fragmentation takes place before the original message is encrypted and authenticated, so that each fragment is separately encrypted and authenticated. On the receiver, the fragments are collected, verified, decrypted, and merged into the original message.

Traffic Selectors for IKEv2

You can configure traffic Selectors in IKEv2 used during IKE negotiation. A traffic selector is an agreement between IKE peers to permit traffic through a VPN tunnel if the traffic matches a specified pair of local and remote addresses. Only the traffic that conforms to a traffic selector is permitted through the associated security association (SA). Traffic selectors are used during the tunnel creation to set up the tunnel and to determine what traffic is allowed through the tunnel.

Proxy ID

A proxy-ID is used during phase 2 of Internet Key Exchange (IKE) Virtual Private Network (VPN) negotiations. Both ends of a VPN tunnel either have a proxy-ID manually configured (route-based VPN) or just use a combination of source IP, destination IP, and service in a tunnel policy. When phase 2 of IKE is negotiated, each end compares the configured local and remote proxy-ID with what is actually received.

Traffic Selectors

Proxy ID is supported for both route-based and policy-based VPNs. However, the multi-proxy ID is supported for only route-based VPNs. The multi-proxy ID is also known as traffic selector. A traffic selector is an agreement between IKE peers to permit traffic through a tunnel, if the traffic matches a specified pair of local and remote addresses. You define a traffic selector within a specific route-based VPN, which can result in multiple Phase 2 IPsec SAs. Only traffic that conforms to a traffic selector is permitted through an SA. The traffic selector is commonly required when remote gateway devices are non-Juniper Networks devices.

IKE Authentication (Preshared Key and Certificate-Based Authentication)

The IKE negotiations provides the ability to establish a secure channel over which two parties can communicate. You can define how the two parties authenticate each other using a preshared key authentication or certificate based authentication.

Preshared Key Authentication	Certificate-Based Authentication
Common way to establish a VPN connection.	Secure way to establish VPN connection.
• Preshared key is a password that is the same for both the parties. This password is exchanged in advance using a phone, through a verbal exchange, or through less secure mechanisms, even e-mail. • Preshared key must consist of at least 8 characters (12 or more is recommended) using a combination of letters, numbers, and nonalphanumeric characters, along with different cases for the letters. • Preshared key must not use a dictionary word.	Certificates are composed of a public and private key, and can be signed by a primary certificate known as a certificate authority (CA)
The parties authenticate each other by encrypting the preshared key with the peer's public key, which is obtained in the Diffie-Hellman exchange.	The parties check certificates to confirm if they are signed by a trusted CA.
Preshared keys are commonly deployed for site-to-site IPsec VPNs, either within a single organization or between different organizations.	Certificates are also far more ideal in larger scale environments with numerous peer sites that should not all share a preshared key.

Network Address Translation-Traversal (NAT-T)

Network Address Translation-Traversal (NAT-T) is a method for getting around IP address translation issues encountered when data protected by IPsec passes through a NAT device for address translation.

Any changes to the IP addressing, which is the function of NAT, causes IKE to discard packets. After detecting one or more NAT devices along the data path during Phase 1 exchanges, NAT-T adds a layer of User Datagram Protocol (UDP) encapsulation to IPsec packets so they are not discarded after address translation. NAT-T encapsulates both IKE and ESP traffic within UDP with port 4500 used as both the source and destination port. Because NAT devices age out stale UDP translations, keepalive messages are required between the peers.

The location of a NAT device can be such that:

- Only the IKEv1 or IKEv2 initiator is behind a NAT device. Multiple initiators can be behind separate NAT devices. Initiators can also connect to the responder through multiple NAT devices.

- Only the IKEv1 or IKEv2 responder is behind a NAT device.

- Both the IKEv1 or IKEv2 initiator and the responder are behind a NAT device.

Suite B and PRIME Cryptographic Suites

Suite B is a set of cryptographic algorithms designated by the U.S. National Security Agency to allow commercial products to protect traffic that is classified at secret or top secret levels. Suite B protocols are defined in RFC 6379, *Suite B Cryptographic Suites for IPsec*. The Suite B cryptographic suites provide Encapsulating Security Payload (ESP) integrity and confidentiality and should be used when ESP integrity protection and encryption are both required. Protocol Requirements for IP Modular Encryption (PRIME), an IPsec profile defined for public sector networks in the United Kingdom, is based on the Suite B cryptographic suite, but uses AES-GCM rather than AES-CBC for IKEv2 negotiations.

The following cryptographic suites are supported:

- Suite-B-GCM-128
  - ESP: Advanced Encryption Standard (AES) encryption with 128-bit keys and 16-octet integrity check value (ICV) in Galois Counter Mode (GCM).
  - IKE: AES encryption with 128-bit keys in cipher block chaining (CBC) mode, integrity using SHA-256 authentication, key establishment using Diffie-Hellman (DH) group 19, and authentication using Elliptic Curve Digital Signature Algorithm (ECDSA) 256-bit elliptic curve signatures.
- Suite-B-GCM-256
  - ESP: AES encryption with 256-bit keys and 16-octet ICV in GCM for ESP.
  - IKE: AES encryption with 256-bit keys in CBC mode, integrity using SHA-384 authentication, key establishment using DH group 20, and authentication using ECDSA 384-bit elliptic curve signatures.
- PRIME-128
  - ESP: AES encryption with 128-bit keys and 16-octet ICV in GCM.
  - IKE: AES encryption with 128-bit keys in GCM, key establishment using DH group 19, and authentication using ECDSA 256-bit elliptic curve signatures.
- PRIME-256

- ESP: AES encryption with 256-bit keys and 16-octet ICV in GCM for ESP.
- IKE: AES encryption with 256-bit keys in GCM, key establishment using DH group 20, and authentication using ECDSA 384-bit elliptic curve signatures.

Suite-B cryptographic suites support IKEv1 and IKEv2. PRIME cryptographic suites only support IKEv2.

IPsec Basics

IN THIS SECTION

IPsec Overview

IN THIS SECTION

IPsec is a suite of related protocols for cryptographically securing communications at the IP Packet Layer. IPsec also provides methods for the manual and automatic negotiation of security associations (SAs) and key distribution, all the attributes for which are gathered in a domain of interpretation (DOI). The IPsec DOI is a document containing definitions for all the security parameters required for the successful negotiation of a VPN tunnel—essentially, all the attributes required for SA and IKE negotiations. See RFC 2407 and RFC 2408 for more information.

To use IPsec security services, you create SAs between hosts. An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec. There are two types of SAs: manual and dynamic.

IPsec supports two modes of security (transport mode and tunnel mode).

Security Associations

A security association (SA) is a unidirectional agreement between the VPN participants regarding the methods and parameters to use in securing a communication channel. Full bidirectional communication requires at least two SAs, one for each direction. Through the SA, an IPsec tunnel can provide the following security functions:

- Privacy (through encryption)
- Content integrity (through data authentication)
- Sender authentication and—if using certificates—nonrepudiation (through data origin authentication)

The security functions you employ depend on your needs. If you need only to authenticate the IP packet source and content integrity, you can authenticate the packet without applying any encryption. On the other hand, if you are concerned only with preserving privacy, you can encrypt the packet without applying any authentication mechanisms. Optionally, you can both encrypt and authenticate the packet. Most network security designers choose to encrypt, authenticate, and replay-protect their VPN traffic.

An IPsec tunnel consists of a pair of unidirectional SAs—one SA for each direction of the tunnel—that specify the security parameter index (SPI), destination IP address, and security protocol (Authentication Header [AH] or Encapsulating Security Payload [ESP] employed. An SA groups together the following components for securing communications:

- Security algorithms and keys.
- Protocol mode, either transport or tunnel. Junos OS devices always use tunnel mode. (See "Packet Processing in Tunnel Mode" on page 170.)
- Key-management method, either manual key or AutoKey IKE.
- SA lifetime.

For inbound traffic, Junos OS looks up the SA by using the following triplet:

- Destination IP address.
- Security protocol, either AH or ESP.
- Security parameter index (SPI) value.

For outbound VPN traffic, the policy invokes the SA associated with the VPN tunnel.

IPsec Key Management

IN THIS SECTION

The distribution and management of keys are critical to using VPNs successfully. Junos OS supports IPsec technology for creating VPN tunnels with three kinds of key creation mechanisms:

- Manual key
- AutoKey IKE with a preshared key or a certificate

You can choose your key creation mechanism—also called authentication method—during Phase 1 and Phase 2 proposal configuration. See "Internet Key Exchange" on page 10.

This topic includes the following sections:

Manual Key

With manual keys, administrators at both ends of a tunnel configure all the security parameters. This is a viable technique for small, static networks where the distribution, maintenance, and tracking of keys are not difficult. However, safely distributing manual-key configurations across great distances poses security issues. Aside from passing the keys face-to-face, you cannot be completely sure that the keys have not been compromised while in transit. Also, whenever you want to change the key, you are faced with the same security issues as when you initially distributed it.

AutoKey IKE

When you need to create and manage numerous tunnels, you need a method that does not require you to configure every element manually. IPsec supports the automated generation and negotiation of keys and security associations using the Internet Key Exchange (IKE) protocol. Junos OS refers to such automated tunnel negotiation as AutoKey IKE and supports AutoKey IKE with preshared keys and AutoKey IKE with certificates.

- AutoKey IKE with preshared keys—Using AutoKey IKE with preshared keys to authenticate the participants in an IKE session, each side must configure and securely exchange the preshared key in advance. In this regard, the issue of secure key distribution is the same as that with manual keys. However, once distributed, an autokey, unlike a manual key, can automatically change its keys at predetermined intervals using the IKE protocol. Frequently changing keys greatly improves security, and automatically doing so greatly reduces key-management responsibilities. However, changing keys increases traffic overhead; therefore, changing keys too often can reduce data transmission efficiency.

A preshared key is a key for both encryption and decryption, which both participants must have before initiating communication.

- AutoKey IKE with certificates—When using certificates to authenticate the participants during an AutoKey IKE negotiation, each side generates a public-private key pair and acquires a certificate. As long as the issuing certificate authority (CA) is trusted by both sides, the participants can retrieve the peer’s public key and verify the peer's signature. There is no need to keep track of the keys and SAs; IKE does it automatically.

Diffie-Hellman Exchange

A Diffie-Hellman (DH) exchange allows participants to produce a shared secret value. The strength of the technique is that it allows participants to create the secret value over an unsecured medium without passing the secret value through the wire. The size of the prime modulus used in each group's calculation differs as shown in the below table. Diffie Hellman (DH) exchange operations can be performed either in software or in hardware. The following Table 1 on page 23 lists different Diffie Hellman (DH) groups and specifies whether the operation performed for that group is in the hardware or in software.

Table 1: Diffie Hellman (DH) groups and their exchange operations performed

Diffie-Hellman (DH) Group	Prime Module Size
DH Group 1	768-bit
DH Group 2	102-bit
DH Group 5	1536-bit
DH Group 14	2048-bit
DH Group 15	3072-bit

Table 1: Diffie Hellman (DH) groups and their exchange operations performed *(Continued)*

Diffie-Hellman (DH) Group	Prime Module Size
DH Group 16	4096-bit
DH Group 19	256-bit elliptic curve
DH Group 20	384-bit elliptic curve
DH Group 21	521-bit elliptic curve
DH Group 24	2048-bit with 256-bit prime order subgroup

Starting in Junos OS Release 19.1R1, SRX Series Firewalls support DH groups 15, 16, and 21.

Starting in Junos OS Release 20.3R1, vSRX Virtual Firewall instances with junos-ike package installed support DH groups 15, 16, and 21.

We do not recommend the use of DH groups 1, 2, and 5.

Because the modulus for each DH group is a different size, the participants must agree to use the same group.

IPsec Security Protocols

IN THIS SECTION

IPsec uses two protocols to secure communications at the IP layer:

- Authentication Header (AH)—A security protocol for authenticating the source of an IP packet and verifying the integrity of its content

- Encapsulating Security Payload (ESP)—A security protocol for encrypting the entire IP packet (and authenticating its content)

You can choose your security protocols—also called *authentication and encryption algorithms*—during Phase 2 proposal configuration. See "Internet Key Exchange" on page 10.

For each VPN tunnel, both AH and ESP tunnel sessions are installed on Services Processing Units (SPUs) and the control plane. Tunnel sessions are updated with the negotiated protocol after negotiation is completed. For SRX5400, SRX5600, and SRX5800 devices, tunnel sessions on anchor SPUs are updated with the negotiated protocol while non-anchor SPUs retain ESP and AH tunnel sessions. ESP and AH tunnel sessions are displayed in the outputs for the `show security flow session` and `show security flow cp-session` operational mode commands.

This topic includes the following sections:

IPsec Authentication Algorithms (AH Protocol)

The Authentication Header (AH) protocol provides a means to verify the authenticity and integrity of the content and origin of a packet. You can authenticate the packet by the checksum calculated through a Hash Message Authentication Code (HMAC) using a secret key and either MD5 or SHA hash functions.

- Message Digest 5 (MD5)—An algorithm that produces a 128-bit hash (also called a *digital signature* or *message digest*) from a message of arbitrary length and a 16-byte key. The resulting hash is used, like a fingerprint of the input, to verify content and source authenticity and integrity.

- Secure Hash Algorithm (SHA)—An algorithm that produces a 160-bit hash from a message of arbitrary length and a 20-byte key. It is generally regarded as more secure than MD5 because of the larger hashes it produces. Because the computational processing is done in the ASIC, the performance cost is negligible.

For more information on MD5 hashing algorithms, see RFC 1321 and RFC 2403. For more information on SHA hashing algorithms, see RFC 2404. For more information on HMAC, see RFC 2104.

IPsec Encryption Algorithms (ESP Protocol)

The Encapsulating Security Payload (ESP) protocol provides a means to ensure privacy (encryption) and source authentication and content integrity (authentication). ESP in tunnel mode encapsulates the entire IP packet (header and payload) and then appends a new IP header to the now-encrypted packet. This new IP header contains the destination address needed to route the protected data through the network. (See "Packet Processing in Tunnel Mode" on page 170.)

With ESP, you can both encrypt and authenticate, encrypt only, or authenticate only. For encryption, you can choose one of the following encryption algorithms:

- Data Encryption Standard (DES)—A cryptographic block algorithm with a 56-bit key.
- Triple DES (3DES)—A more powerful version of DES in which the original DES algorithm is applied in three rounds, using a 168-bit key. DES provides significant performance savings but is considered unacceptable for many classified or sensitive material transfers.
- Advanced Encryption Standard (AES)—An encryption standard which offers greater interoperability with other devices. Junos OS supports AES with 128-bit, 192-bit, and 256-bit keys.

For authentication, you can use either MD5 or SHA algorithms.

Even though it is possible to select NULL for encryption, it has been demonstrated that IPsec might be vulnerable to attack under such circumstances. Therefore, we suggest that you choose an encryption algorithm for maximum security.

IPsec Tunnel Negotiation

The following two different modes that determine how the traffic is exchanged in the VPN.

- Tunnel mode—Protect traffic by encapsulating the original IP packet within another packet in the VPN tunnel. This mode uses preshared keys with IKE to authenticate peers or digital certificates with IKE to authenticate peers. This is most commonly used when hosts within separate private networks want to communicate over a public network. This mode can be used by both VPN clients and VPN gateways, and protects communications that come from or go to non-IPsec systems.
- Transport mode—Protect traffic by sending the packet directly between the two hosts that have established the IPsec tunnel. That is, when the communication endpoint and cryptographic endpoint are the same. The data portion of the IP packet is encrypted, but the IP header is not. VPN gateways that provide encryption and decryption services for protected hosts cannot use transport mode for protected VPN communications. The IP addresses of the source or destination can be modified if the packet is intercepted. Because of its construction, transport mode can be used only when the communication endpoint and cryptographic endpoint are the same.

Supported IPsec and IKE Standards

On routers equipped with one or more MS-MPCs, MS-MICs, or DPCs, the Canada and U.S. version of Junos OS substantially supports the following RFCs, which define standards for IP Security (IPsec) and Internet Key Exchange (IKE).

- RFC 2085, *HMAC-MD5 IP Authentication with Replay Prevention*

- RFC 2401, *Security Architecture for the Internet Protocol* (obsoleted by RFC 4301)
- RFC 2402, *IP Authentication Header* (obsoleted by RFC 4302)
- RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH*
- RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH* (obsoleted by RFC 4305)
- RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*
- RFC 2406, *IP Encapsulating Security Payload (ESP)* (obsoleted by RFC 4303 and RFC 4305)
- RFC 2407, *The Internet IP Security Domain of Interpretation for ISAKMP* (obsoleted by RFC 4306)
- RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)* (obsoleted by RFC 4306)
- RFC 2409, *The Internet Key Exchange (IKE)* (obsoleted by RFC 4306)
- RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*
- RFC 2451, *The ESP CBC-Mode Cipher Algorithms*
- RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*
- RFC 3193, *Securing L2TP using IPsec*
- RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*
- RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*
- RFC 3948, *UDP Encapsulation of IPsec ESP Packets*
- RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)*
- RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*
- RFC 4211, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*
- RFC 4301, *Security Architecture for the Internet Protocol*
- RFC 4302, *IP Authentication Header*
- RFC 4303, *IP Encapsulating Security Payload (ESP)*
- RFC 4305, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*
- RFC 4306, *Internet Key Exchange (IKEv2) Protocol*

- RFC 4307, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*
- RFC 4308, *Cryptographic Suites for IPsec*

  Only Suite VPN-A is supported in Junos OS.
- RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*
- RFC 4835, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*
- RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)* (obsoleted by RFC 7296)
- RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)*
- RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*

Junos OS partially supports the following RFCs for IPsec and IKE:

- RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)*
- RFC 5114, *Additional Diffie-Hellman Groups for Use with IETF Standards*
- RFC 5903, *Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2*

The following RFCs and Internet draft do not define standards, but provide information about IPsec, IKE, and related technologies. The IETF classifies them as “Informational.”

- RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*
- RFC 2412, *The OAKLEY Key Determination Protocol*
- RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*
- Internet draft draft-eastlake-sha2-02.txt, *US Secure Hash Algorithms (SHA and HMAC-SHA)* (expires July 2006)

SEE ALSO

Services Interfaces Overview for Routing Devices

MX Series 5G Universal Routing Platform Interface Module Reference

Accessing Standards Documents on the Internet

Release History Table

Release	Description
19.1R1	Starting in Junos OS Release 19.1R1, SRX Series Firewalls support DH groups 15, 16, and 21.

3

CHAPTER

PKI in Junos OS

PKI in Junos OS

SUMMARY

This topic describes the basic elements of public key infrastructure (PKI) in Junos OS.

IN THIS SECTION

A public key infrastructure (PKI) supports the distribution and identification of public encryption keys, enabling users to both securely exchange data over networks such as the Internet and verify the identity of the other party.

Introduction to PKI in Junos OS

IN THIS SECTION

PKI Applications Overview

The Junos OS uses public/private keys in the following areas:

- SSH/SCP (for secure command-line interface [CLI]-based administration)
- Secure Sockets Layer (SSL) (for secure Web-based administration and for https-based webauth for user authentication)
- Internet Key Exchange (IKE) (for IPsec VPN tunnels)

NOTE: Note the following points:

- Currently Junos OS supports only IKE (using public key infrastructure (PKI) certificates for public key validation).
- The SSH and SCP are used exclusively for system administration and depends on the use of out-of-band fingerprints for public key identity binding and validation. Details on SSH are not covered in this topic.

Components for Administering PKI in Junos OS

The following components are required for administrating PKI in Junos OS:

- CA certificates and authority configuration
- Local certificates including the devices identity (example: IKE ID type and value) and private and public keys
- Certificate validation through a certificate revocation list (CRL)

Basic Elements of PKI in Junos OS

Junos OS supports three specific types of PKI objects:

- Private/public key pair
- Certificates
  - Local certificate—The local certificate contains the public key and identity information for the Juniper Networks device. The Juniper Networks device owns the associated private key. This certificate is generated based on a certificate request from the Juniper Networks device.
  - Pending certificate — A pending certificate contains a key pair and identity information that is generated into a PKCS10 certificate request and manually sent to a certificate authority (CA). While the Juniper Networks device waits for the certificate from the CA, the existing object (key pair and the certificate request) is tagged as a certificate request or pending certificate.

    NOTE: Junos OS Release 9.0 and later supports automatic sending of certificate requests through SCEP.

  - CA certificate — When the certificate is issued by the CA and loaded into the Junos OS device, the pending certificate is replaced by the newly generated local certificate. All other certificates loaded into the device are considered CA certificates.

- Certificate revocation lists (CRLs)

Note the following points about certificates:

- Local certificates are generally used when a Junos OS device has VPNs in more than one administrative domain.
- All PKI objects are stored in a separate partition of persistent memory, apart from the Junos OS image and the system's general configuration.
- Each PKI object has a unique name or certificate-ID given to it when it is created and maintains that ID until its deletion. You can view the certificate-ID by using the `show security pki local-certificate` command.
- A certificate cannot be copied from a device under most circumstances. The private key on a device must be generated on that device only, and it should never be viewed or saved from that device. So PKCS12 files (which contain a certificate with the public key and the associated private key) are not supported on Junos OS devices.
- CA certificates validate the certificates received by the IKE peer. If the certificate is valid, then it is verified in the CRL to see whether the certificate has been revoked.

  Each CA certificate includes a CA profile configuration that stores the following information:

  - CA identity, which is typically the domain name of the CA
  - E-mail address for sending the certificate requests directly to the CA
  - Revocation settings:
    - Revocation check enable/disable option
    - Disabling of revocation check in case of CRL download failure.
    - Location of CRL Distribution Point (CDP) (for manual URL setting)
    - CRL refresh interval

PKI Components In Junos OS

IN THIS SECTION

This topic includes the following sections:

PKI Management and Implementation

The minimum PKI elements required for certificate-based authentication in Junos OS are:

- CA certificates and authority configuration.
- Local certificates including the device's identity (example: IKE ID type and value) and private and public keys
- Certificate validation through a CRL.

Junos OS supports three different types of PKI objects:

Internet Key Exchange

The procedure for digitally signing messages sent between two participants in an Internet Key Exchange (IKE) session is similar to digital certificate verification, with the following differences:

- Instead of making a digest from the CA certificate, the sender makes it from the data in the IP packet payload.
- Instead of using the CA's public-private key pair, the participants use the sender's public-private key pair.

Trusted CA Group

A Certificate Authority (CA) is a trusted third party responsible for issuing and revoking certificates. You can group multiple CAs (CA profiles) in one trusted CA group for a given topology. These certificates are used to establish connection between two endpoints. To establish IKE or IPsec, both the endpoints must trust the same CA. If either of the endpoints are unable to validate the certificate using their respective trusted CA (ca-profile) or trusted CA group, the connection is not established.

For example, there are two endpoints, endpoint A and endpoint B are trying to establish a secure connection. When endpoint B presents it's certificate to endpoint A, the endpoint A will check if the certificate is valid. The CA of the endpoint A verifies the signed certificate that the endpoint B is using to get authorized. When `trusted-ca` or `trusted-ca-group` is configured, the device will only use the CA profiles added in this `trusted-ca-group` or the CA profile configured under `trusted-ca` to validate the certificate coming from endpoint B. If the certificate is verified as valid, the connection is allowed, else the connection is rejected.

Benefits:

- For any incoming connection request, only the certificate issued by that particular trusted CA of that endpoint gets validated. If not, the authorization will reject establishing the connection.

Cryptographic Key Handling Overview

With cryptographic key handling, persistent keys are stored in the memory of the device without any attempt to alter them. While the internal memory device is not directly accessible to a potential adversary, those who require a second layer of defense can enable special handling for cryptographic keys. When enabled, the cryptographic key handling encrypts keys when not immediately in use, performs error detection when copying a key from one memory location to another, and overwrites the memory location of a key with a random bit pattern when the key is no longer in use. Keys are also protected when they are stored in the flash memory of the device. Enabling cryptographic key handling feature does not cause any externally observable change in the behavior of the device, and the device continues to interoperate with the other devices.

A cryptographic administrator can enable and disable the cryptographic self-test functions; however, the security administrator can modify the behavior of the cryptographic self-test functions such configuring periodic self-tests or selecting a subset of cryptographic self-tests.

The following persistent keys are currently under the management of IKE and PKI:

- IKE preshared keys (IKE PSKs)

- PKI private keys

- Manual VPN keys

SEE ALSO

IKE Authentication (Certificate-Based Authentication) | 144

IPsec Overview | 20

request security pki generate-key-pair (Security) | 1739

RELATED DOCUMENTATION

Certificate Enrollment | 51

Digital Certificates

IN THIS SECTION

A digital certificate is an electronic means for verifying your identity through a trusted third party, known as a certificate authority (CA). Alternatively, you can use a self-signed certificate to attest to your identity.

Manual certificate processing includes generation of a PKCS10 request, submission to the CA, retrieval of the signed certificate, and manually loading the certificate into the Juniper Networks device. Based on your deployment environment, you can use either SCEP or CMPv2 for online certificate enrollment.

To use a digital certificate to authenticate your identity when establishing a secure VPN connection, you must first do the following:

- Obtain a CA certificate from which you intend to obtain a local certificate, and then load the CA certificate onto the device. The CA certificate can contain a CRL to identify invalid certificates.
- Obtain a local certificate from the CA whose CA certificate you have previously loaded, and then load the local certificate in the device. The local certificate establishes the identity of the Juniper Networks device with each tunnel connection.

Manually Generating Digital Certificates: Configuration Overview

To obtain digital certificates manually:

1. Generate a key pair on the device. See "Self-Signed Digital Certificates" on page 37.
2. Create a CA profile or profiles containing information specific to a CA. See "Example: Configuring a CA Profile" on page 47.
3. Generate the CSR for the local certificate and send it to the CA server. See "Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server" on page 62.
4. Load the certificate onto the device. See "Example: Loading CA and Local Certificates Manually" on page 63.

5. Configure automatic reenrollment. See "Example: Using SCEP to Automatically Renew a Local Certificate" on page 56.

6. If necessary, load the certificate's CRL on the device. See "Example: Manually Loading a CRL onto the Device" on page 70.

7. If necessary, configure the CA profile with CRL locations. See "Example: Configuring a Certificate Authority Profile with CRL Locations" on page 75

Self-Signed Digital Certificates

IN THIS SECTION

A self-signed certificate is a certificate that is signed by the same entity who created it rather than by a Certificate Authority (CA). Junos OS provides two methods for generating a self-signed certificate-automatic generation and manual generation.

Understanding Self-Signed Certificates

A self-signed certificate is a certificate that is signed by its creator rather than by a Certificate Authority (CA).

Self-signed certificates allow for use of SSL-based (Secure Sockets Layer) services without requiring that the user or administrator to undertake the considerable task of obtaining an identity certificate signed by a CA.

Self-signed certificates do not provide additional security as do those generated by CAs. This is because a client cannot verify that the server he or she has connected to is the one advertised in the certificate.

Junos OS provides two methods for generating a self-signed certificate:

- Automatic generation

  In this case, the creator of the certificate is the Juniper Networks device. An automatically generated self-signed certificate is configured on the device by default.

  After the device is initialized, it checks for the presence of an automatically generated self-signed certificate. If it does not find one, the device generates one and saves it in the file system.

- Manual generation

  In this case, you create the self-signed certificate for the device.

  At any time, you can use the CLI to generate a self-signed certificate. These certificates are also used to gain access to SSL services.

Self-signed certificates are valid for five years from the time they were generated.

An automatically generated self-signed certificate allows for use of SSL-based services without requiring that the administrator obtain an identity certificate signed by a CA.

A self-signed certificate that is automatically generated by the device is similar to a Secure Shell (SSH) host key. It is stored in the file system, not as part of the configuration. It persists when the device is rebooted, and it is preserved when a `request system snapshot` command is issued.

A self-signed certificate that you manually generate allows for use of SSL-based services without requiring that you obtain an identity certificate signed by a CA. A manually generated self-signed certificate is one example of a public key infrastructure (PKI) local certificate. As is true of all PKI local certificates, manually generated self-signed certificates are stored in the file system.

SEE ALSO

PKI in Junos OS | 31

Example: Generating a Public-Private Key Pair

IN THIS SECTION

This example shows how to generate a public-private key pair.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

In this example, you generate a public-private key pair named ca-ipsec.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To generate a public-private key pair:

- Create a certificate key pair.

```
user@host> request security pki generate-key-pair certificate-id ca-ipsec
```

Verification

After the public-private key pair is generated, the Juniper Networks device displays the following:

```
generated key pair ca-ipsec, key size 1024 bits
```

SEE ALSO

Example: Verifying Certificate Validity | 77

Example: Manually Generating Self-Signed Certificates

IN THIS SECTION

This example shows how to generate self-signed certificates manually.

Requirements

Before you begin, generate a public private key pair. See "Digital Certificates" on page 36.

Overview

For a manually generated self-signed certificate, you specify the DN when you create it. For an automatically generated self-signed certificate, the system supplies the DN, identifying itself as the creator.

In this example, you generate a self-signed certificate with the e-mail address as `mholmes@example.net`. You specify a certificate-id of `self-cert` to be referenced by web management, which refers a Example: Generating a Public-Private Key Pair-pair of the same certificate-id.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To generate the self-signed certificate manually:

1. Create the self-signed certificate.

```
user@host> request security pki local-certificate generate-self-signed certificate-id self-
cert subject CN=abc  domain-name example.net ip-address 1.2.3.4 email mholmes@example.net
```

Verification

To verify the certificate was properly generated and loaded, enter the `show security pki local-certificate` operational mode command.

Using Automatically Generated Self-Signed Certificates (CLI Procedure)

After the device is initialized, it checks for the presence of a self-signed certificate. If a self-signed certificate is not present, the device automatically generates one.

You can add the following statement to your configuration if you want to use the automatically generated self-signed certificate to provide access to HTTPS services:

```
system {
    services {
        web-management {
            http {
                interface [ ... ];
            } https {
                system-generated-certificate;
                interface [ ... ];
            }
        }
    }
}
```

The device uses the following distinguished name for the automatically generated certificate:

```
“ CN=<device serial number>, CN=system generated, CN=self-signed”
```

Use the following command to specify that the automatically generated self-signed certificate is to be used for Web management HTTPS services:

```
user@host# set system services web-management https system-generated-certificate
```

Use the following operational command to delete the automatically generated self-signed certificate:

```
user@host# clear security pki local-certificate system-generated
```

After you delete the system-generated self-signed certificate, the device automatically generates a new one and saves it in the file system.

RELATED DOCUMENTATION

PKI in Junos OS | **31**

Certificate Authority

IN THIS SECTION

A certificate authority (CA) profile define every parameter associated with a specific certificate to establish secure connection between two endpoints. The profiles specify which certificates to use, how to verify certificate revocation status, and how that status constrains access.

Configuring a Trusted CA Group

IN THIS SECTION

This section describes the procedure to create a trusted CA group for a list of CA profiles and delete a trusted CA group.

Creating a Trusted CA Group for a List of CA Profiles

You can configure and assign a trusted CA group to authorize an entity. When a peer tries to establish a connection with a client, only the certificate issued by that particular trusted CA of that entity gets validated. The device validates if the issuer of the certificate and the one presenting the certificate belongs to the same client network. If the issuer and the presenter belong to the same client network then the connection is established. If not, the connection will not be established.

Before you begin, you must have a list of all the CA profiles you want to add to the trusted group.

In this example, we are creating three CA profiles named `orgA-ca-profile`, `orgB-ca-profile`, and `orgC-ca-profile` and associating the following CA identifiers `ca-profile1`, `ca-profile2`, and `ca-profile3` for the respective profiles. You can group all the three CA profiles to belong to a trusted CA group `orgABC-trusted-ca-group`.

You can configure a maximum of 20 CA profiles for a trusted CA group.

1. Create CA profiles and associate CA identifiers to the profile.

```
[edit]
user@host# set security pki ca-profile orgA-ca-profile ca-identity ca-profile1
user@host# set security pki ca-profile orgB-ca-profile ca-identity ca-profile2
user@host# set security pki ca-profile orgC-ca-profile ca-identity ca-profile3
```

2. Group the CA profiles under a trusted CA group.

```
[edit]
set security pki trusted-ca-group orgABC-trusted-ca-group ca-profiles [orgA-ca-profile orgB-
ca-profile orgC-ca-profile]]
```

3. Commit the configuration when you are done configuring the CA profiles and the trusted CA groups.

```
[edit]
user@host# commit
```

To view the CA profiles and the trusted CA groups configured on your device, run `show security pki` command.

```
user@host# show security pki
ca-profile orgA-ca-profile {
    ca-identity ca-profile1;
}
ca-profile orgB-ca-profile {
    ca-identity ca-profile2;
}
ca-profile orgC-ca-profile {
    ca-identity ca-profile3;
}
trusted-ca-group orgABC-trusted-ca-group {
    ca-profiles [ orgA-ca-profile orgB-ca-profile orgC-ca-profile ];
}
```

The `show security pki` command displays all the CA profiles that are grouped under the `orgABC_trusted-ca-group`.

Deleting a CA Profile from a Trusted CA Group

You can delete a specific CA profile in a trusted CA group or you can delete the trusted CA group itself.

For example, if you want to delete a CA profile named `orgC-ca-profile` from a trusted CA group `orgABC-trusted-ca-group`, configured on your device as shown in "Configuring a Trusted CA Group" on page 43 topic perform the following steps:

1. Delete a CA profile from the trusted CA group.

```
[edit]
user@host# delete security pki trusted-ca-group orgABC-trusted-ca-group ca-profiles orgC-ca-
profile
```

2. If you are done deleting the CA profile from the trusted CA group, commit the configuration.

```
[edit]
user@host# commit
```

To view the `orgC-ca-profile` being deleted from the `orgABC-trusted-ca-group` , run the `show security pki` command.

```
user@host# show security pki
ca-profile orgA-ca-profile {
    ca-identity ca-profile1;
}
ca-profile orgB-ca-profile {
    ca-identity ca-profile2;
}
trusted-ca-group orgABC-trusted-ca-group {
    ca-profiles [ orgA-ca-profile orgB-ca-profile ];
}
```

The output does not display the `orgC-ca-profile` profile as it is deleted from the trusted CA group.

Deleting a Trusted CA Group

An entity can support many trusted CA groups and you can delete any trusted CA group for an entity.

For example, if you want to delete a trusted CA group named `orgABC-trusted-ca-group`, configured on your device as shown in "Configuring a Trusted CA Group" on page 43 topic perform the following steps:

1. Delete a trusted CA group.

```
[edit]
user@host# delete security pki trusted-ca-group orgABC-trusted-ca-group
```

2. If you are done deleting the CA profile from the trusted CA group, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

To view the orgABC-trusted-ca-group being deleted from the entity , run the show security pki command.

```
user@host# show security pki
ca-profile orgA-ca-profile {
    ca-identity ca-profile1;
}
ca-profile orgB-ca-profile {
    ca-identity ca-profile2;
}
```

The output does not display the orgABC-trusted-ca-group as it is deleted from the entity.

RELATED DOCUMENTATION

Understanding Certificate Authority Profiles | 46

Understanding Certificate Authority Profiles

A certificate authority (CA) profile configuration contains information specific to a CA. You can have multiple CA profiles on an SRX Series Firewall. For example, you might have one profile for orgA and one for orgB. Each profile is associated with a CA certificate. If you want to load a new CA certificate without removing the older one then create a new CA profile (for example, Microsoft-2008).

Starting with Junos OS Release 18.1R1, the CA server can be an IPv6 CA server.

The PKI module supports IPv6 address format to enable the use of SRX Series Firewalls in networks where IPv6 is the only protocol used.

A CA issues digital certificates, which helps to establish secure connection between two endpoints through certificate validation. You can group multiple CA profiles in one trusted CA group for a given topology. These certificates are used to establish a connection between two endpoints. To establish IKE or IPsec, both the endpoints must trust the same CA. If either of the endpoints are unable to validate the certificate using their respective trusted CA (ca-profile) or trusted CA group, the connection is not established. A minimum of one CA profile is mandatory to create a trusted CA group and maximum of

20 CAs are allowed in one trusted CA group. Any CA from a particular group can validate the certificate for that particular endpoint.

Starting with Junos OS Release 18.1R1, validation of a configured IKE peer can be done with a specified CA server or group of CA servers. A group of trusted CA servers can be created with the `trusted-ca-group` configuration statement at the `[edit security pki]` hierarchy level; one or multiple CA profiles can be specified. The trusted CA server is bound to the IKE policy configuration for the peer at `[edit security ike policy` *`policy`* `certificate]` hierarchy level.

If proxy profile is configured in CA profile, the device connects to the proxy host instead of the CA server while certificate enrollment, verification or revocation. The proxy host communicates with the CA server with the requests from the device, and then relay the response to the device.

CA proxy profile supports SCEP, CMPv2, and OCSP protocols.

CA proxy profile is supported only on HTTP and is not supported on HTTPS protocol.

SEE ALSO

PKI Components In Junos OS | 33

Example: Configuring a CA Profile

IN THIS SECTION

This example shows how to configure a CA profile.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

In this example, you create a CA profile called `ca-profile-ipsec` with CA identity microsoft-2008. You then create proxy profile to the CA profile. The configuration specifies that the CRL be refreshed every 48 hours, and the location to retrieve the CRL is `http://www.my-ca.com`. Within the example, you set the enrollment retry value to 20. (The default retry value is 10.)

Automatic certificate polling is set to every 30 minutes. If you configure retry only without configuring a retry interval, then the default retry interval is 900 seconds (or 15 minutes). If you do not configure retry or a retry interval, then there is no polling.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To configure a CA profile:

1. Create a CA profile.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec ca-identity microsoft-2008
user@host#
```

2. Optionally, configure the proxy profile to the CA profile.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec proxy-profile px-profile
```

Public key infrastructure (PKI) uses proxy profile configured at the system-level. The proxy profile being used in the CA profile must be configured at the `[edit services proxy]` hierarchy. There can be more than one proxy profile configured under `[edit services proxy]` hierarchy. Each CA profile is referred to the most one such proxy profile. You can configure host and port of the proxy profile at the `[edit system services proxy]` hierarchy.

3. Create a revocation check to specify a method for checking certificate revocation.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec ca-identity microsoft-2008 revocation-check crl
```

4. Set the refresh interval, in hours, to specify the frequency in which to update the CRL. The default values are next-update time in CRL, or 1 week, if no next-update time is specified.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec ca-identity microsoft-2008 revocation-check crl refresh-interval 48 url http://www.my-ca.com/my-crl.crl
```

5. Specify the enrollment retry value.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec enrollment retry 20
```

6. Specify the time interval in seconds between attempts to automatically enroll the CA certificate online.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec enrollment retry-interval 1800
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the `show security pki` command.

Example: Configuring an IPv6 address as the Source Address for a CA Profile

This example shows how to configure an IPv6 address as the source address for a CA profile.

No special configuration beyond device initialization is required before configuring this feature.

In this example, create a CA profile called `orgA-ca-profile` with CA identity `v6-ca` and set the source address of the CA profile to be an IPv6 address, such as `2001:db8:0:f101::1`. You can configure the enrollment URL to accept an IPv6 address `http://[2002:db8:0:f101::1]:/.../`.

1. Create a CA profile.

```
[edit]
user@host# set security pki ca-profile orgA-ca-profile ca-identity v6_ca
```

2. Configure the source address of the CA profile to be an IPv6 address.

```
[edit]
user@host# set security pki ca-profile v6_ca source-address 2001:db8:0:f101::1
```

3. Specify the enrollment parameters for the CA.

```
[edit]
user@host# set security pki ca-profile v6_ca enrollment url http://[2002:db8:0:f101::1]:/.../
```

4. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

SEE ALSO

Example: Configuring a Certificate Authority Profile with CRL Locations | 75

Release History Table

Release	Description
18.1R1	Starting with Junos OS Release 18.1R1, the CA server can be an IPv6 CA server.
18.1R1	Starting with Junos OS Release 18.1R1, validation of a configured IKE peer can be done with a specified CA server or group of CA servers.

RELATED DOCUMENTATION

Self-Signed Digital Certificates | 37

Certificate Enrollment

IN THIS SECTION

A certificate authority (CA) issues digital certificates, which helps to establish a secure connection between two endpoints through certificate validation. The following topics describe how to configure CA certificates online or local using Simple Certificate Enrollment Protocol (SCEP):

Enrolling Digital Certificates Online: Configuration Overview

You can use either Certificate Management Protocol version 2 (CMPv2) or Simple Certificate Enrollment Protocol (SCEP) to enroll digital certificates. To enroll a certificate online:

1. Generate a key pair on the device. See "Self-Signed Digital Certificates" on page 37.
2. Create a CA profile or profiles containing information specific to a CA. See "Example: Configuring a CA Profile" on page 47.
3. For SCEP only, enroll the CA certificate. See "Enrolling a CA Certificate Online Using SCEP" on page 53.
4. Enroll the local certificate from the CA whose CA certificate you have previously loaded. See "Example: Enrolling a Local Certificate Online Using SCEP" on page 54.
5. Configure automatic reenrollment. See "Example: Using SCEP to Automatically Renew a Local Certificate" on page 56.

Understanding Online CA Certificate Enrollment

With Simple Certificate Enrollment Protocol (SCEP), you can configure your Juniper Networks device to obtain a certificate authority (CA) certificate online and start the online enrollment for the specified certificate ID. The CA public key verifies certificates from remote peers.

Understanding Local Certificate Requests

When you create a local certificate request, the device generates an end entity certificate in PKCS #10 format from a key pair you previously generated using the same certificate ID.

A subject name is associated with the local certificate request in the form of a common name (CN), organizational unit (OU), organization (O), locality (L), state (ST), country (C), and domain component (DC). Additionally, a subject alternative name is associated in the following form:

- IP address
- E-mail address
- Fully qualified domain name (FQDN)

Specify the subject name in the distinguished name format in quotation marks, including the domain component (DC), common name (CN), serial number (SN), organizational unit name (OU), organization name (O), locality (L), state (ST), and country (C).

Some CAs do not support an e-mail address as the domain name in a certificate. If you do not include an e-mail address in the local certificate request, you cannot use an e-mail address as the local IKE ID when configuring the device as a dynamic peer. Instead, you can use a fully qualified domain name (if it is in the local certificate), or you can leave the local ID field empty. If you do not specify a local ID for a dynamic peer, enter the *hostname.domain-name* of that peer on the device at the other end of the IPsec tunnel in the peer ID field.

Enrolling a CA Certificate Online Using SCEP

Before you begin:

1. Generate a public and private key pair. See "Self-Signed Digital Certificates" on page 37.

2. Create a CA profile. See "Example: Configuring a CA Profile" on page 47.

To enroll a CA certificate online:

1. Retrieve the CA certificate online using SCEP. (The attributes required to reach the CA server are obtained from the defined CA profile.)

   ```
   user@host> request security pki ca-certificate enroll ca-profile ca-profile-ipsec
   ```

   The command is processed synchronously to provide the fingerprint of the received CA certificate.

   ```
   Fingerprint:
   e6:fa:d6:da:e8:8d:d3:00:e8:59:12:e1:2c:b9:3c:c0:9d:6c:8f:8d (sha1)
   82:e2:dc:ea:48:4c:08:9a:fd:b5:24:b0:db:c3:ba:59 (md5)
   Do you want to load the above CA certificate ? [yes,no]
   ```

2. Confirm that the correct certificate is loaded. The CA certificate is loaded only when you type **yes** at the CLI prompt.

   For more information on the certificate, such as the bit length of the key pair, use the command `show security pki ca-certificate`.

Example: Enrolling a Local Certificate Online Using SCEP

IN THIS SECTION

This example shows how to enroll a local certificate online using Simple Certificate Enrollment Protocol (SCEP).

Requirements

Before you begin:

- Generate a public and private key pair. See "Self-Signed Digital Certificates" on page 37.
- Configure a certificate authority profile. See "Example: Configuring a CA Profile" on page 47.
- For SCEP, enroll the CA certificate. See "Enrolling a CA Certificate Online Using SCEP" on page 53.

Overview

In this example, you configure your Juniper Networks device to obtain a local certificate online and start the online enrollment for the specified certificate ID with SCEP. You specify the URL path to the CA server in the CA profile name `ca-profile-ipsec`.

You use the `request security pki local-certificate enroll scep` command to start the online enrollment for the specified certificate ID. (Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the `scep` keyword is supported and required.) You must specify the CA profile name (for example, `ca-profile-ipsec`), the certificate ID corresponding to a previously generated key-pair (for example, `qqq`), and the following information:

- The challenge password provided by the CA administrator for certificate enrollment and reenrollment.
- At least one of the following values:
  - The domain name to identify the certificate owner in IKE negotiations—for example, `qqq.example.net`.

- The identity of the certificate owner for IKE negotiation with the e-mail statement—for example, `qqq@example.net`.
- The IP address if the device is configured for a static IP address—for example, `10.10.10.10`.

Specify the subject name in the distinguished name format in quotation marks, including the domain component (DC), common name (CN), serial number (SN), organizational unit name (OU), organization name (O), locality (L), state (ST), and country (C).

Once the device certificate is obtained and the online enrollment begins for the certificate ID. The command is processed asynchronously.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To enroll a local certificate online:

1. Specify the CA profile.

```
[edit]
user@host# set security pki ca-profile ca-profile-ipsec  enrollment url path-to-ca-server
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

3. Initiate the enrollment process by running the operational mode command.

```
user@host> request security pki local-certificate enroll scep ca-profile ca-profile-ipsec
certificate-id qqq challenge-password ca-provided-password domain-name qqq.example.net email
```

```
qqq@example.net ip-address 10.10.10.10 subject DC=example, CN=router3, SN, OU=marketing,
O=example, L=sunnyvale, ST=california, C=us
```

If you define SN in the subject field without the serial number, then the serial number is read directly from the device and added to the certificate signing request (CSR).

Starting in Junos OS Release 19.4R2, a warning message `ECDSA Keypair not supported with SCEP for cert_id` *`<certificate id>`* is displayed when you try to enroll local certificate using an Elliptic Curve Digital Signature Algorithm (ECDSA) key with Simple Certificate Enrollment Protocol (SCEP) as ECDSA key is not supported with SCEP.

Verification

To verify the configuration is working properly, enter the `show security pki` command.

Example: Using SCEP to Automatically Renew a Local Certificate

IN THIS SECTION

You can use either Certificate Management Protocol version 2 (CMPv2) or Simple Certificate Enrollment Protocol (SCEP) to enroll digital certificates. This example shows how to renew the local certificates automatically using SCEP.

Requirements

Before you begin:

- Obtain a certificate either on line or manually. See "Digital Certificates" on page 36.
- Obtain a local certificate. See "Example: Enrolling a Local Certificate Online Using SCEP" on page 54.

Overview

You can enable the device to automatically renew certificates that were acquired by online enrollment or loaded manually. Automatic certificate renewal saves you from having to remember to renew certificates on the device before they expire, and helps to maintain valid certificates at all times.

Automatic certificate renewal is disabled by default. You can enable automatic certificate renewal and configure the device to automatically send out a request to reenroll a certificate before it expires. You can specify when the certificate reenrollment request is to be sent; the trigger for reenrollment is the percentage of the certificate's lifetime that remains before expiration. For example, if the renewal request is to be sent when the certificate's remaining lifetime is 10 percent, then configure 10 for the reenrollment trigger.

For this feature to work, the device must be able to reach the CA server, and the certificate must be present on the device during the renewal process. Furthermore, you must also ensure that the CA issuing the certificate can return the same DN. The CA must not modify the subject name or alternate subject name extension in the new certificate.

You can enable and disable automatic SCEP certificate renewal either for all SCEP certificates or on a per-certificate basis. You use the `set security pki auto-re-enrollment scep` command to enable and configure certificate reenrollment. In this example, you specify the certificate ID of the CA certificate as `ca-ipsec` and set the CA profile name associated with the certificate to `ca-profile-ipsec`. You set the challenge password for the CA certificate to the challenge password provided by the CA administrator; this password must be the same one configured previously for the CA. You also set the percentage for the reenrollment trigger to `10`. During automatic reenrollment, the Juniper Networks device by default uses the existing key pair. A good security practice is to regenerate a new key pair for reenrollment. To generate a new key pair, use the `re-generate-keypair` command.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To enable and configure local certificate reenrollment:

1. To enable and configure certificate reenrollment.

```
[edit]
user@host# set security pki auto-re-enrollment scep certificate-id ca-ipsec  ca-profile-name
ca-profile-ipsec  challenge-password ca-provided-password re-enroll-trigger-time-percentage
10 re-generate-keypair
```

   Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the `scep` keyword is supported and required.

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the `show security pki local-certificate detail` operational mode command.

Understanding CMPv2 and SCEP Certificate Enrollment

Based on your deployment environment, you can use either Certificate Management Protocol version 2 (CMPv2) or Simple Certificate Enrollment Protocol (SCEP) for online certificate enrollment. This topic describes some of the basic differences between the two protocols.

Table 2 on page 58 describes the differences between the CMPv2 and SCEP certificate enrollment protocols.

Table 2: Comparison of CMPv2 and SCEP Certificate Enrollment

Attribute	CMPv2	SCEP
Supported certificate types:	DSA, ECDSA, and RSA	RSA only
Supported standards	RFCs 4210 and 4211	Internet Engineering Task Force draft

Certificate enrollment and reenrollment requests and responses differ between CMPv2 and SCEP. With CMPv2, there is no separate command to enroll CA certificates. With SCEP, you enroll CA certificates with the `request security pki ca-certificate enroll` command and specify the CA profile. A CA profile must be configured with either CMPv2 or SCEP.

Understanding Certificate Enrollment with CMPv2

IN THIS SECTION

The `request security pki local-certificate enroll cmpv2` command uses CMPv2 to enroll a local digital certificate online. This command loads both end-entity and CA certificates based on the CA server configuration. The CA profile must be created prior to CA certificate enrollment because the enrollment URL is extracted from the CA profile.

This topic describes certificate enrollment with the CMPv2 protocol.

Certificate Enrollment and Reenrollment Messages

The CMPv2 protocol mainly involves certificate enrollment and reenrollment operations. The certificate enrollment process includes Initialization Request and Initialization Response messages, while certificate reenrollment includes Key Update Request and Key Update Response messages.

CMPv2 server responds back with Initialization Response (IP). The response contains end-entity certificate along with optional CA certificates. The message integrity and message authenticity of Initialization Response can be verified using shared-secret-information according to RFC 4210. The Initialization Response can also be verified using issuer CA public-key. Before you re-enroll an end-entity certificate, you must have a valid CA certificate enrolled on the device.

The Initialization Response or Key Update Response message can contain an issuer CA certificate or a chain of CA certificates. The CA certificates received in the responses are treated as trusted CA certificates and stored in the receiving device if they are not already present in the trusted CA store. These CA certificates are later used for end-entity certificate validation.

We do not support CA certificate re-enrollment. If a CA certificate expires, you must unenroll the current CA certificate and enroll it back again.

End-Entity Certificate with Issuer CA Certificate

In a simple scenario, the Initialization Response message might contain only an end-entity certificate, in which case the CA information is provided separately. The certificate is stored in the end-entity certificate store.

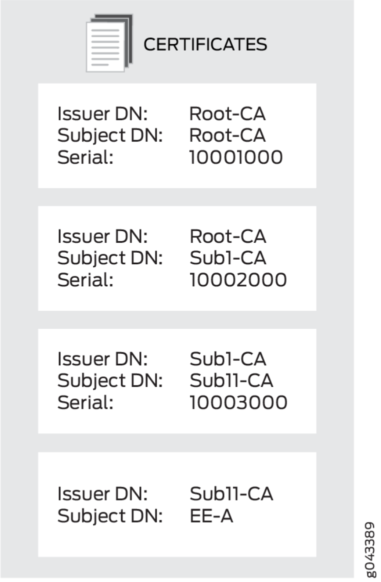
The Initialization Response message can contain an end-entity certificate as well as a self-signed issuer CA certificate. The end-entity certificate is first stored in the certificate store, and then the CA certificate is checked. If the CA certificate is found and the subject distinguished name (DN) of the CA certificate in the Initialization Response message matches the issuer DN of the end-entity certificate, the CA certificate is stored in the CA certificate store for the CA profile name specified in the CMPv2 certificate enrollment command. If the CA certificate already exists in the CA certificate store, no action is taken.

End-Entity Certificate with CA Certificate Chain

In many deployments, the end-entity certificate is issued by an intermediate CA in a certificate chain. In this case, the Initialization Response message can contain the end-entity certificate along with a list of CA certificates in the chain. The intermediate CA certificates and the self-signed root CA certificates are all required to validate the end-entity certificate. The CA chain might also be needed to validate certificates received from peer devices with similar hierarchies. The following section describes how certificates in the CA chain are stored.

In Figure 4 on page 61, the Initialization Response message includes the end-entity certificate and three CA certificates in a certificate chain.

Figure 4: End-Entity Certificate with CA Certificate Chain



The end-entity certificate is stored in the end-entity certificate store. Each CA certificate needs a CA profile. The CA certificate with the subject DN Sub11-CA is the first CA in the chain and is the issuer of the end-entity certificate. It is stored in the CA profile that is specified with the CMPv2 certificate enrollment command.

Each of the remaining CA certificates in the chain is checked for its presence in the CA store. If a CA certificate is not present in the CA store, it is saved and a CA profile is created for it. The new CA profile name is created using the least significant 16 digits of the CA certificate serial number. If the serial number is longer than 16 digits, the most significant digits beyond 16 digits are truncated. If the serial number is shorter than 16 digits, the remaining most significant digits are filled with `0`s. For example, if the serial number is 11111000100010001000, then the CA profile name is `1000100010001000`. If the serial number is 10001000, then the CA profile name is `0000000010001000`.

It is possible that multiple certificate serial numbers can have the same least significant 16 digits. In that case, `-00` is appended to the profile name to create a unique CA profile name; additional CA profile names are created by incrementing the appended number, from `-01` up to `-99`. For example, CA profile names can be `1000100010001000`, `1000100010001000-00`, and `1000100010001000-01`.

SEE ALSO

Understanding Certificate Authority Profiles | 46

IKE Authentication (Certificate-Based Authentication)

Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server

IN THIS SECTION

This example shows how to generate a certificate signing request manually.

Requirements

Generate a public and private key. See "Self-Signed Digital Certificates" on page 37.

Overview

In this example, you generate a certificate request using the certificate ID of a public-private key pair you previously generated (ca-ipsec). Then you specify the domain name (example.net) and the associated common name (abc). The certificate request is displayed in PEM format.

You copy the generated certificate request and paste it into the appropriate field at the CA website to obtain a local certificate. (Refer to the CA server documentation to determine where to paste the certificate request.) When the PKCS #10 content is displayed, the MD5 hash and SHA-1 hash of the PKCS #10 file is also displayed.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To generate a local certificate manually:

- Specify certificate ID, domain name, and common name.

```
user@host> request security pki generate-certificate-request certificate-id ca-ipsec domain-name example.net subject CN=abc
```

Verification

To view the certificate signing request, enter the `show security pki certificate-request detail` command.

```
Certificate identifier: ca-ipsec
Certificate version: 1
Issued to: CN = abc
Public key algorithm: rsaEncryption(1024 bits)
30:81:89:02:81:81:00:da:ea:cd:3a:49:1f:b7:33:3c:c5:50:fb:57
de:17:34:1c:51:9b:7b:1c:e9:1c:74:86:69:a4:36:77:13:a7:10:0e
52:f4:2b:52:39:07:15:3f:39:f5:49:d6:86:70:4b:a6:2d:73:b6:68
39:d3:6b:f3:11:67:ee:b4:40:5b:f4:de:a9:a4:0e:11:14:3f:96:84
03:3c:73:c7:75:f5:c4:c2:3f:5b:94:e6:24:aa:e8:2c:54:e6:b5:42
c7:72:1b:25:ca:f3:b9:fa:7f:41:82:6e:76:8b:e6:d7:d2:93:9b:38
fe:fd:71:01:2c:9b:5e:98:3f:0c:ed:a9:2b:a7:fb:02:03:01:00:01
Fingerprint:
0f:e6:2e:fc:6d:52:5d:47:6e:10:1c:ad:a0:8a:4c:b7:cc:97:c6:01 (sha1)
f8:e6:88:53:52:c2:09:43:b7:43:9c:7a:a2:70:98:56 (md5)
```

Example: Loading CA and Local Certificates Manually

IN THIS SECTION

This example shows how to load CA and local certificates manually.

Requirements

Before you begin:

- Generate a public-private key pair. See "Self-Signed Digital Certificates" on page 37.
- Create a CA profile. See "Understanding Certificate Authority Profiles" on page 46.

  CA Profile is only required for the CA certificate and not for the local certificate
- Generate a certificate request. See "Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server" on page 62.

Overview

In this example, you download the local.cert and ca.cert certificates and save them to the /var/tmp/ directory on the device.

After you download certificates from a CA, you transfer them to the device (for example, using FTP), and then load them.

You can load the following certificate files onto a device running Junos OS:

- A local or end-entity (EE) certificate that identifies your local device. This certificate is your public key.
- A CA certificate that contains the CA's public key.
- A CRL that lists any certificates revoked by the CA.

  You can load multiple EE certificates onto the device.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To load the certificate files onto a device:

1. Load the local certificate.

   ```
   [edit]
   user@host> request security pki local-certificate load certificate-id local.cert
   filename /var/tmp/local.cert
   ```

2. Load the CA certificate.

   ```
   [edit]
   user@host> request security pki ca-certificate load ca-profile ca-profile-ipsec
   filename /var/tmp/ca.cert
   ```

3. Examine the fingerprint of the CA certificate, if it is correct for this CA certificate select yes to accept.

Verification

To verify the certificates loaded properly, enter the `show security pki local-certificate` and `show security pki ca-certificate` commands in operational mode.

```
Fingerprint:
e8:bf:81:6a:cd:26:ad:41:b3:84:55:d9:10:c4:a3:cc:c5:70:f0:7f (sha1)
19:b0:f8:36:e1:80:2c:30:a7:31:79:69:99:b7:56:9c (md5)
Do you want to load this CA certificate ? [yes,no] (no) yes
```

SEE ALSO

Example: Using SCEP to Automatically Renew a Local Certificate | 56

Example: Verifying Certificate Validity | 77

Example: Configuring a Certificate Authority Profile with CRL Locations | 75

Deleting Certificates (CLI Procedure)

You can delete a local or trusted CA certificate that is automatically or manually generated.

Use the following command to delete a local certificate:

```
user@host> clear security pki local certificate certificate-id (certificate-id| all | system-
generated )
```

Specify a certificate ID to delete a local certificate with a specific ID, use `all` to delete all local certificates, or specify `system-generated` to delete the automatically generated self-signed certificate.

When you delete an automatically generated self-signed certificate, the device generates a new one.

To delete a CA certificate:

```
user@host> clear security pki ca-certificate ca-profile (ca-profile-name | all)
```

Specify a CA profile to delete a specific CA certificate, or use `all` to delete all CA certificates present in the persistent store.

You are asked for confirmation before a CA certificate can be deleted.

Release History Table

Release	Description
19.4R2	Starting in Junos OS Release 19.4R2, a warning message `ECDSA Keypair not supported with SCEP for cert_id <certificate id>` is displayed when you try to enroll local certificate using an Elliptic Curve Digital Signature Algorithm (ECDSA) key with Simple Certificate Enrollment Protocol (SCEP) as ECDSA key is not supported with SCEP.
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the `scep` keyword is supported and required.
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the `scep` keyword is supported and required.

Certificate Revocation

IN THIS SECTION

Digital certificates have an expiration date, however, prior to expiration, a certificate may no longer be valid due to many reasons. You can manage certificate revocations and validations locally and by referencing a Certificate Authority (CA) certificate revocation list (CRL).

Understanding Online Certificate Status Protocol and Certificate Revocation Lists

IN THIS SECTION

OCSP is used to check the revocation status of X509 certificates. OCSP provides revocation status on certificates in real time and is useful in time-sensitive situations such as bank transactions and stock trades.

The revocation status of a certificate is checked by sending a request to an OCSP server that resides outside of an SRX Series Firewall. Based on the response from the server, the VPN connection is allowed or denied. OCSP responses are not cached on SRX Series Firewalls.

The OCSP server can be the *certificate authority (CA)* that issues a certificate or a designated authorized responder. The location of the OCSP server can be configured manually or extracted from the certificate that is being verified. Requests are sent first to OCSP server locations that are manually configured in CA profiles with the `ocsp url` statement at the [`edit security pki ca-profile` *`profile-name`* `revocation-check`] hierarchy level; up to two locations can be configured for each CA profile. If the first configured OCSP server is not reachable, the request is sent to the second OCSP server. If the second OCSP server is not reachable, the request is then sent to the location in the certificate's AuthorityInfoAccess extension field. The `use-ocsp` option must also be configured, as *certificate revocation list (CRL)* is the default checking method.

SRX Series Firewalls accept only signed OCSP responses from the CA or authorized responder. The response received is validated using trusted certificates. The response is validated as follows:

1. The CA certificate enrolled for the configured CA profile is used to validate the response.

2. The OCSP response might contain a certificate to validate the OCSP response. The received certificate must be signed by a CA certificate enrolled in the SRX Series Firewall. After the received certificate is validated by the CA certificate, it is used to validate the OCSP response.

The response from the OCSP server can be signed by different CAs. The following scenarios are supported:

- The CA server that issues the end entity certificate for a device also signs the OCSP revocation status response. The SRX Series Firewall verifies the OCSP response signature using the CA certificate enrolled in the SRX Series Firewall. After the OCSP response is validated, the certificate revocation status is checked.

- An authorized responder signs the OCSP revocation status response. The certificate for the authorized responder and the end entity certificate being verified must be issued by the same CA. The authorized responder is first verified using the CA certificate enrolled in the SRX Series Firewall. The OCSP response is validated using the responder’s CA certificate. The SRX Series Firewall then uses the OCSP response to check the revocation status of the end entity certificate.

- There are different CA signers for the end entity certificate being verified and the OCSP response. The OCSP response is signed by a CA in the certificate chain for the end entity certificate being verified. (All peers participating in an IKE negotiation need to have at least one common trusted CA in their respective certificate chains.) The OCSP responder’s CA is verified using a CA in the certificate chain. After validating the responder CA certificate, the OCSP response is validated using the responder’s CA certificate.

To prevent replay attacks, a *nonce* payload can be sent in an OCSP request. Nonce payloads are sent by default unless it is explicitly disabled. If enabled, the SRX Series Firewall expects the OCSP response to contain a nonce payload, otherwise the revocation check fails. If OCSP responders are not capable of responding with a nonce payload, then the nonce payload must be disabled on the SRX Series Firewall.

In the normal course of business, certificates are revoked for various reasons. You might wish to revoke a certificate if you suspect that it has been compromised, for example, or when a certificate holder leaves the company.

You can manage certificate revocations and validations in two ways:

- Locally— This is a limited solution.
- By referencing a Certificate Authority (CA) certificate revocation list (CRL)— You can automatically access the CRL online at intervals you specify or at the default interval set by the CA.

In Phase 1 negotiations, SRX Series Firewall verifies the EE certificate received from the peer during an IKE exchange and uses the CRL to make sure the EE certificate is not revoked by its CA.

If a CRL is not loaded on the device and the peer certificate issuer is a trusted CA:

1. Junos OS retrieves the CRL through the configured LDAP or HTTP CRL locations (that is, the CRL Distribution Points (CDP)), if they are defined in the CA profile.
2. If the CRL Distribution Points is not configured in the CA profile, the device uses the CDP extension in a certificate issued by the CA (if present). The certificate issued by the CA can be a certificate enrolled by the administrator or received during the Phase 1 negotiation.

If the EE certificate is not issued by a root CA, the certificates of each intermediate CAs goes through the same verification and revocation check. The CRL of the root CA is used to check if the certificate issued by the root CA is revoked. If the CDP is not configured in the root CA profile, the device uses the CDP extension in the certificate issued by the CA (if present).

The CRL distribution point extension (.cdp) in an X509 certificate can be added to either an HTTP URL or an LDAP URL.

If the certificate does not contain a certificate distribution point extension, and you cannot automatically retrieve the CRL through Lightweight Directory Access Protocol (LDAP) or Hypertext Transfer Protocol (HTTP), you can retrieve a CRL manually and load that in the device.

Local certificates are being validated against certificate revocation list (CRL) even when CRL check is disabled. This can be stopped by disabling the CRL check through the Public Key Infrastructure (PKI) configuration. When CRL check is disabled, PKI will not validate local certificate against CRL.

Comparison of Online Certificate Status Protocol and Certificate Revocation List

Online Certificate Status Protocol (OCSP) and certificate revocation list (CRL) can both be used to check the revocation status of a certificate. There are advantages and disadvantages to each method.

- OCSP provides certificate status in real time, while CRL uses cached data. For time-sensitive applications, OCSP is the preferred approach.

- CRL checking is faster because lookup for certificate status is done on information cached on the VPN device. OCSP requires time to obtain the revocation status from an external server.

- CRL requires additional memory to store the revocation list received from a CRL server. OCSP does not require additional memory to save the revocation status of certificates.

- OCSP requires that the OCSP server be available at all times. CRL can use cached data to check the revocation status of certificates when the server is unreachable.

On MX Series and SRX Series Firewalls, CRL is the default method used to check the revocation status of a certificate.

SEE ALSO

Certificate Validation

Example: Manually Loading a CRL onto the Device

IN THIS SECTION

This example shows how to load a CRL manually onto the device.

Requirements

Before you begin:

1. Generate a public and private key pair. See "Self-Signed Digital Certificates" on page 37.

2. Generate a certificate request. See "Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server" on page 62.

3. Configure a certificate authority (CA) profile. See "Example: Configuring a CA Profile" on page 47.

4. Load your certificate onto the device. See "Example: Loading CA and Local Certificates Manually" on page 63.

Overview

You can load a CRL manually, or you can have the device load it automatically, when you verify certificate validity. To load a CRL manually, you obtain the CRL from a CA and transfer it to the device (for example, using FTP).

In this example, you load a CRL certificate called `revoke.crl` from the /var/tmp directory on the device. The CA profile is called `ca-profile-ipsec`. (Maximum file size is 5 MB.)

If a CRL is already loaded into the ca-profile the command `clear security pki crl ca-profile ca-profile-ipsec` must be run first to clear the old CRL.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To load a CRL certificate manually:

1. Load a CRL certificate.

```
[edit]
user@host> request security pki crl load ca-profile ca-profile-ipsec filename /var/tmp/
revoke.crl
```

Junos OS supports loading of CA certificates in X509, PKCS #7, DER, or PEM formats.

Verification

To verify the configuration is working properly, enter the `show security pki crl` operational mode command.

Understanding Dynamic CRL Download and Checking

Digital certificates are issued for a set period of time and are invalid after the specified expiration date. A CA can revoke an issued certificate by listing it in a certificate revocation list (CRL). During peer certificate validation, the revocation status of a peer certificate is checked by downloading the CRL from a CA server to the local device.

To facilitate the CRL check for the certificates when a CA profile is not configured, dynamic CA profile is created. A dynamic CA profile is automatically created on the local device with the format `dynamic-`*`nnn`*.

A dynamic CA profile:

- Allows the local device to download the Dynamic CA and Dynamic CRL (for corresponding CA) as per peer’s localcert issuer

- Checks the revocation status of the peer's certificate

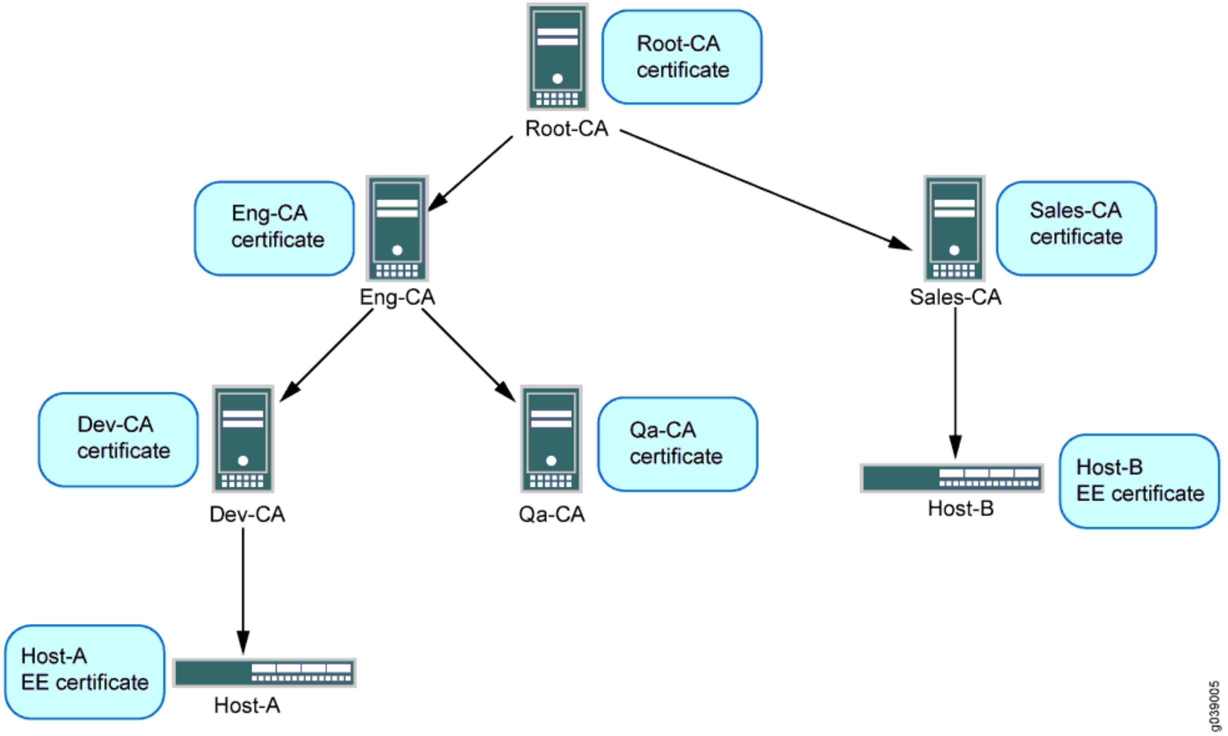
A VPN device checks a peer's EE certificate for its revocation status. A VPN device uses the certificate received from its peer to do the following:

- Extract the URL to dynamically download the CA's CRL

- Check the revocation status of the peer's EE certificate

In Figure 5 on page 73, Host-A can use the Sales-CA and EE certificates received from Host-B to dynamically download the CRL for Sales-CA and check the revocation status of Host-B’s certificate.

Figure 5: Multilevel Hierarchy for Certificate-Based Authentication



In case of single hierarchy CA servers or CA certificate chain, the local EE certificate and the received peer EE certificate are issued from the same CA server.

Following are some of the SRX Series Firewall behavior based on different configurations:

- If you have configured a SRX Series Firewall with a trusted-ca or trusted-ca-group, then the device does not validate or trust any other CAs.

- If you have defined a CA profile that has a chain of CAs where the SRX Series Firewall only trusts the root CA and peer has a certificate signed by a sub-CA to this root, then Dynamic CA and CRL will be added to the device.

Table 3 on page 74 provides few sample scenarios where Dynamic CA or CRL is not created:

Table 3: Sample Scenarios

Scenario	Condition
Sample scenario 1	In the CA profile, you have defined a trusted CA for ca-profile-name, and you receive a connection from a device that has a certificate signed by a different CA that was not defined as a trusted CA in your CA profile.
Sample scenario 2	You have defined a CA profile that has a chain of CAs where the SRX Series Firewall only trust a sub-CA, and peer has a certificate signed by a level above this sub-CA.

To enable dynamic CA profiles, you must configure the `revocation-check crl` option on a Root-CA profile at the [`edit security pki ca-profile` *`profile-name`*] hierarchy level.

The revocation check properties of a Root-CA profile are inherited for dynamic CA profiles. In Figure 5 on page 73, the CA profile configuration on Host-A for Root-CA enables dynamic CA profiles as shown in the following output:

```
admin@host-A# show security
pki {
    ca-profile Root-CA {
        ca-identity Root-CA;
        enrollment {
            url “www.example.net/scep/Root/”;
        }
        revocation-check {
            crl;
        }
    }
}
```

A dynamic CA profile is created on Host-A for Sales-CA. Revocation checking is inherited for the Sales-CA dynamic CA profile from Root-CA.

If the `revocation-check disable` statement is configured in a Root-CA profile, dynamic CA profiles are not created and dynamic CRL download and checking is not performed.

The data for CRLs downloaded from dynamic CA profiles are displayed with the `show security pki crl` command in the same way as CRLs downloaded by configured CA profiles. The CRL from a dynamic CA profile is updated periodically as are those for CA profiles that are configured in the device. The peer CA certificate is also required for signature validation of CRL downloaded from CA server.

The CA certificate is required to validate the CRL received from a CA server; therefore, the CA certificate received from a peer is stored on the local device. The received CA certificate from peer is used to validate the CRL and the certificate it issued. Because the received CA certificate is not enrolled by an administrator, the result of a successful certificate verification is not conclusive until the whole certificate chain up to the root CA is verified. The certificate of the root CA must be enrolled by an administrator.

SEE ALSO

PKI in Junos OS | 31

Configuring a Trusted CA Group

Example: Configuring a Device for Peer Certificate Chain Validation | 147

Understanding Certificate Authority Profiles | 46

Example: Configuring a Certificate Authority Profile with CRL Locations

IN THIS SECTION

This example shows how to configure a certificate authority profile with CRL locations.

Requirements

Before you begin:

1. Generate a key pair in the device. See "Digital Certificates" on page 36.

2. Create a CA profile or profiles containing information specific to a CA. See "Example: Configuring a CA Profile" on page 47.

3. Obtain a personal certificate from the CA. See "Example: Manually Generating a CSR for the Local Certificate and Sending It to the CA Server" on page 62.

4. Load the certificate onto the device. See "Example: Loading CA and Local Certificates Manually" on page 63.

5. Configure automatic reenrollment. See Example: Configuring SecurID User Authentication.

6. If necessary, load the certificate's CRL on the device. See "Example: Manually Loading a CRL onto the Device" on page 70.

Overview

In this example, you direct the device to check the validity of the CA profile called `my_profile` and, if a CRL did not accompany a CA certificate and is not loaded on the device, to retrieve the CRL from the URL `http://abc/abc-crl.crl`.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To configure certificate using CRL:

1. Specify the CA profile and URL.

```
[edit]
user@host# set security pki ca-profile my_profile revocation-check crl url http://abc/abc-
crl.crl
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the `show security pki` operational mode command.

SEE ALSO

Deleting Certificates (CLI Procedure) | 66

Example: Verifying Certificate Validity

IN THIS SECTION

This example shows how to verify the validity of a certificate.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

In this example, you verify certificates manually to find out whether a certificate has been revoked or whether the CA certificate used to create a local certificate is no longer present on the device.

When you verify certificates manually, the device uses the CA certificate (`ca-cert`) to verify the local certificate (`local.cert`). If the local certificate is valid, and if `revocation-check` is enabled in the CA profile, the device verifies that the CRL is loaded and valid. If the CRL is not loaded and valid, the device downloads the new CRL.

For CA-issued certificates or CA certificates, a DNS must be configured in the device's configuration. The DNS must be able to resolve the host in the distribution CRL and in the CA cert/revocation list url in the ca-profile configuration. Additionally, you must have network reachability to the same host in order for the checks to receive.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To manually verify the validity of a certificate:

1. Verify the validity of a local certificate.

```
[edit]
user@host> request security pki local-certificate verify certificate-id local.cert
```

2. Verify the validity of a CA certificate.

```
[edit]
user@host> request security pki ca-certificate verify ca-profile ca-profile-ipsec
```

The associated private key and the signature are also verified.

Verification

To verify the configuration is working properly, enter the `show security pki ca-profile` command.

If an error is returned instead of a positive verification the failure is logged in pkid.

SEE ALSO

Deleting Certificates (CLI Procedure) | 66

Deleting a Loaded CRL (CLI Procedure)

You can choose to delete a loaded CRL if you no longer need to use it to manage certificate revocations and validation.

Use the following command to delete a loaded certificate revocation list:

```
user@host> clear security pki crl ca-profile (ca-profile all)
```

Specify a CA profile to delete a CRL associated with the CA identified by the profile, or use `all` to delete all CRLs.

SEE ALSO

Deleting Certificates (CLI Procedure) | 66

RELATED DOCUMENTATION

Certificate Authority | 42

Certificate Validation

IN THIS SECTION

Understanding Digital Certificate Validation

IN THIS SECTION

During IKE negotiation, the PKI daemon on an SRX Series Firewall validates X509 certificates received from VPN peers. The certificate validation performed is specified in RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. Basic certificate and certificate chain validations include signature and date validation as well as revocation checks. This topic describes additional digital certificate validations performed by the PKI daemon.

Policy Validation

X509 certificates can include optional policy validation fields. If a policy validation field is present, policy validation is performed for the entire certificate chain including the end entity (EE) certificate and intermediate certificate authority (CA) certificates. Policy validation is not applicable to the root certificate. Policy validation ensures that the EE and intermediate CA certificates have a common policy. If no common policy exists for the certificate chain being validated, certificate validation fails.

Prior to policy validation, a certificate chain containing the self-signed root certificate, intermediate CA certificates, and EE certificate must be built. The policy validation starts with the intermediate CA certificate issued by the self-signed root certificate and continues through the EE certificate.

The following optional certificate fields are used for policy validation:

- **policy-oids**
- **requireExplicitPolicy**
- **skipCerts**

These fields are described in the following sections.

Policy OIDs Configured on SRX Series Firewalls

In some situations, it might be desirable to only accept certificates with known policy object identifiers (OIDs) from peers. This optional configuration allows certificate validation to succeed only if the certificate chain received from the peer contains at least one policy OID that is configured on the SRX Series Firewall.

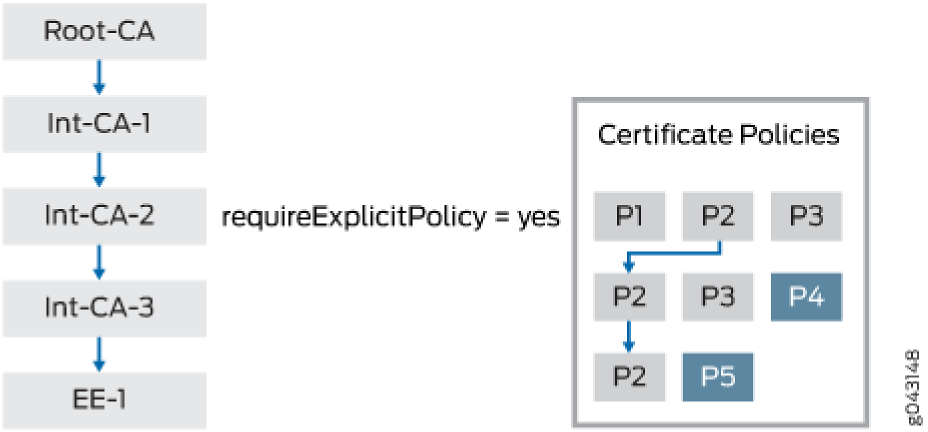
On the SRX Series Firewall, policy OIDs are configured in an IKE policy with the `policy-oids` configuration statement at the [`edit security ike policy` *`policy-name`* `certificate`] hierarchy level. You can configure up to five policy OIDs. For a peer's certificate to be validated successfully, the peer's certificate chain must contain at least one of the policy OIDs configured on the SRX Series Firewall. Note that the **policy-oids** field in a certificate is optional. If you configure policy OIDs on the SRX Series Firewall but the peer's certificate chain does not contain any policy OIDs, certificate validation fails.

No Policy OIDs Configured on SRX Series Firewalls

If no policy OID is configured on the SRX Series Firewall, policy validation starts whenever the **requireExplicitPolicy** field is encountered in the certificate chain. A certificate can contain one or more certificate policy OIDs. For policy validation to succeed, there must be a common policy OID in the certificate chain.

Figure 6 on page 82 shows a certificate chain that consists of certificates for a root CA, three intermediate CAs, and an EE. The CA certificate for Int-CA-2 contains the **requireExplicitPolicy** field; therefore, policy validation starts with Int-CA-2 and continues through EE-1. The certificate for Int-CA-2 contains policy OIDs P1, P2, and P3. The certificate for Int-CA-3 contains policy OIDs P2, P3, and P4. The certificate for EE-1 contains policy OIDs P2 and P5. Because the policy OID P2 is common to the certificates being validated, policy validation succeeds.

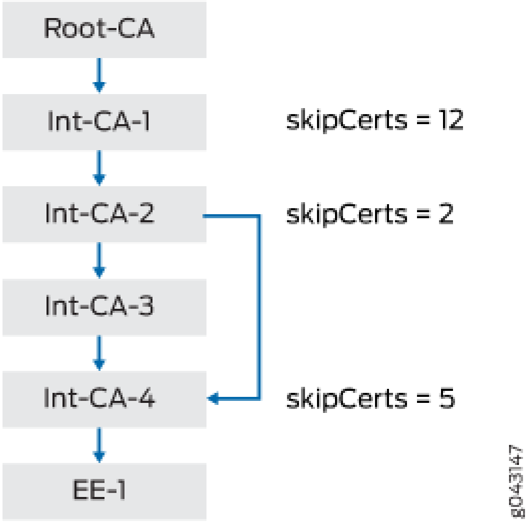
Figure 6: Policy Validation with requireExplicitPolicy Field



The optional **skipCerts** field in an intermediate CA certificate indicates the number of certificates, including the current CA certificate, that are to be excluded from policy validation. If **skipCerts** is 0, policy validation starts from the current certificate. If **skipCerts** is 1, the current certificate is excluded from policy validation. The value of the **skipCerts** field is checked in every intermediate CA certificate. If a **skipCerts** value is encountered that is lower than the current number of certificates being excluded, the lower **skipCerts** value is used.

Figure 7 on page 83 shows a certificate chain consisting of a root CA, four intermediate CAs, and an EE. The **skipCerts** value in Int-CA-1 is 12, which skips 12 certificates including the certificate for Int-CA-1. However, the **skipCerts** value is checked in every intermediate CA certificate in the chain. The **skipCerts** value in Int-CA-2 is 2, which is lower than 12, so now 2 certificates are skipped. The **skipCerts** value in Int-CA-4 is 5, which is greater than 2, so the Int-CA-4 **skipCerts** value is ignored.

Figure 7: Policy Validation with skipCerts Field



When policy OIDs are configured on the SRX Series Firewall, the certificate fields **requireExplicitPolicy** and **skipCerts** are ignored.

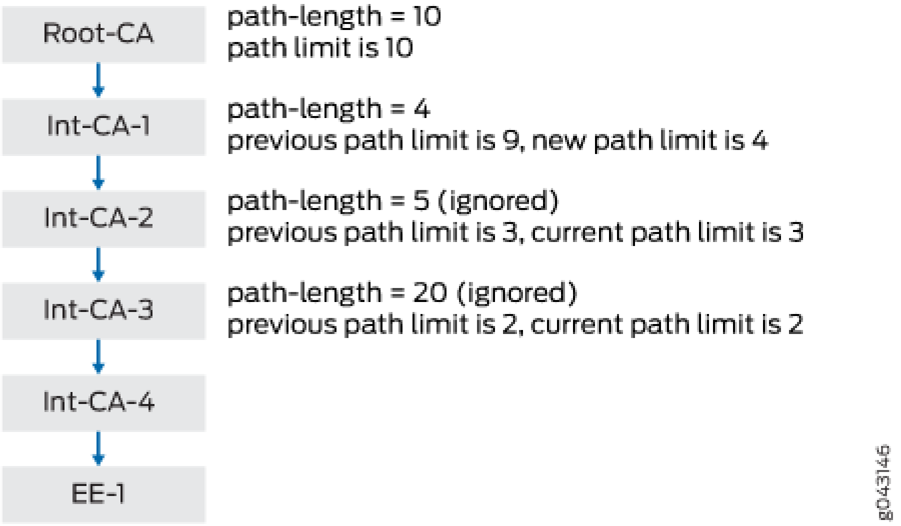
Path Length Validation

Certificate validation can involve a certificate chain that includes a root CA, one or more optional intermediate CAs, and an EE certificate. The number of intermediate CAs can grow depending upon the deployment scenario. Path length validation provides a mechanism to limit the number of intermediate certificates involved in certificate validation. **path-length** is an optional field in an X509 certificate. The value of **path-length** indicates the number of non-self-signed intermediate CA certificates allowed for certificate validation. The last certificate, which is generally the EE certificate, is not included in the path limit. If the root certificate contains a **path-length** value of 0, no intermediate CA certificates are allowed. If the **path-length** value is 1, there can be 0 or 1 intermediate CA certificates.

path-length can be present in multiple CA certificates in the certificate chain. The path length validation always begins with the self-signed root certificate. The path limit is decremented by 1 at each intermediate certificate in the chain. If an intermediate certificate contains a **path-length** value less than the current path limit, the new limit is enforced. On the other hand, if the **path-length** value is larger than the current path limit, it is ignored.

Figure 8 on page 84 shows a certificate chain that consists of a root CA, four intermediate CAs, and an EE. The **path-length** value in Root-CA is 10, therefore the initial path limit of non-self-signed intermediate CA certificates allowed for certificate validation is 10. At Int-CA-1, the path limit is 10-1 or 9. The **path-length** value in Int-CA-1 is 4, which is less than the path limit of 9, so the new path limit becomes 4. At Int-CA-2, the path limit is 4-1 or 3. The **path-length** value in Int-CA-2 is 5, which is larger than the path limit of 3, so it is ignored. At Int-CA-3, the path limit is 3-1 or 2. The **path-length** value in Int-CA-3 is 20, which is larger than the path limit of 2, so it is also ignored.

Figure 8: Path Length Validation


Root-CA
path-length = 10
path limit is 10
Int-CA-1
path-length = 4
previous path limit is 9, new path limit is 4
Int-CA-2
path-length = 5 (ignored)
previous path limit is 3, current path limit is 3
Int-CA-3
path-length = 20 (ignored)
previous path limit is 2, current path limit is 2
Int-CA-4
EE-1
g043146

Key Usage

The key usage field in an EE or CA certificate defines the purpose of the key contained in the certificate.

- For EE certificates, if the key usage field is present but the certificate does not contain **digitalSignature** or **nonrepudiation** flags, the certificate is rejected. If the key usage field is not present, then key usage is not checked.

- For CA certificates, the key can be used for certificate or CRL signature validation. Because the PKI daemon is responsible for both X509 certificate validation and CRL downloads, key usage must be checked before validating the certificate or CRL.

  In certificate signature validation, the **keyCertSign** flag indicates that a CA certificate can be used for certificate signature validation. If this flag is not set, certificate validation is terminated.

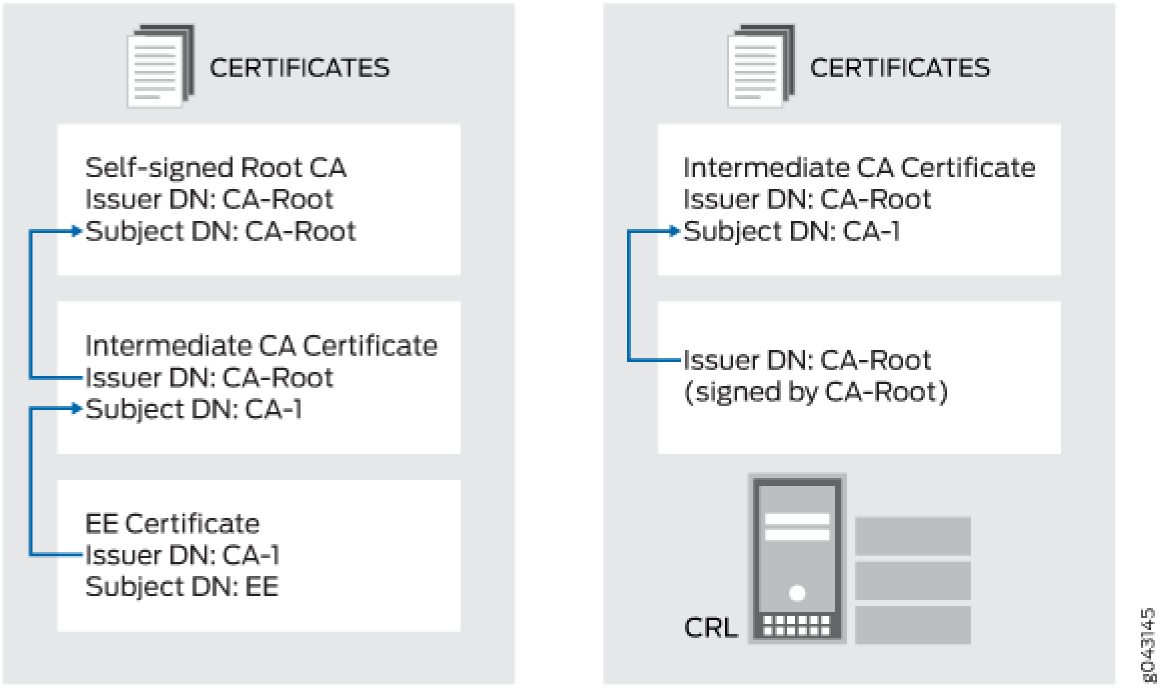
  In Phase 1 negotiations of CRL signature validation, participants check the certificate revocation list (CRL) to see if certificates received during an IKE exchange are still valid. The CRL is periodically downloaded for CA profiles configured with CRL as the certificate revocation check. Downloaded CRL files must be verified before they are downloaded into the device. One of the verification steps is to validate the CRL signature using a CA certificate. The downloaded CRL is signed with the CA certificate's private key and it must be verified with the CA certificate's public key stored in the device. The key usage field in the CA certificate must contain the **CRLSign** flag to verify the downloaded CRL. If this flag is not present, the CRL is discarded.

Issuer and Subject Distinguished Name Validation

Signature validation is performed for certificates received from a peer as well as for the CRL file downloaded from a CA server. Signature validation involves looking up the CA certificate in a CA database based on the issuer's distinguished name (DN) in the certificate or the CRL being verified.

Figure 9 on page 85 shows the lookup for CA certificates based on the issuer DN. In the EE certificate, the issuer DN is CA-1, which is the subject DN of the intermediate CA certificate in the chain. In the intermediate CA certificate, the issuer DN is CA-Root, which is the subject DN of the self-signed Root-CA certificate in the chain. In the CRL, the issuer DN is CA-Root, which is the subject DN of the self-signed Root-CA certificate.

Figure 9: Issuer and Subject DN Validation



The lookup for the issuer or subject DN must follow these rules for attribute values:

- Attribute values encoded in different ASN.1 types (for example, PrintableString and BMPString) are assumed to represent different strings.

- Attribute values encoded in PrintableString types are not case-sensitive. These attribute values are compared after removing leading and trailing white spaces and converting internal substrings of one or more consecutive white spaces to a single space.
- Attribute values encoded in types other than PrintableString are case-sensitive.

SEE ALSO

Example: Configuring a Certificate Authority Profile with CRL Locations | 75
Deleting Certificates (CLI Procedure) | 66

Example: Validating Digital Certificate by Configuring Policy OIDs on an SRX Series Firewall

IN THIS SECTION

In some situations, it might be desirable to only accept certificates with known policy object identifiers (OIDs) from peers. This optional configuration allows certificate validation to succeed only if the certificate chain received from the peer contains at least one policy OID that is configured on the SRX Series Firewall. This example shows how to configure policy OIDs in the IKE policy on an SRX Series Firewall.

You must ensure that at least one of the policy OIDs configured on the SRX Series Firewall is included in a peer's certificate or certificate chain. Note that the **policy-oids** field in a peer's certificate is optional. If you configure policy OIDs in an IKE policy and the peer's certificate chain does not contain any policy OIDs, certificate validation for the peer fails.

Requirements

Before you begin:

- Ensure that you are using Junos OS Release 12.3X48-D10 or later for SRX Series Firewalls.
- Configure an IPsec VPN tunnel. See "IPsec VPN with Autokey IKE Configuration Overview" on page 190. The complete IKE phase 1 and phase 2 VPN tunnel configuration is not shown in this example.

Overview

This example shows an IKE policy configuration where policy OIDs 2.16.840.1.101.3.1.48.2 and 5.16.40.1.101.3.1.55.2 are specified. The IKE policy ike_cert_pol references the IKE proposal ike_cert_prop, which is not shown. The local certificate on the SRX Series Firewall is lc-igloo-root.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike policy ike_cert_pol mode main
set security ike policy ike_cert_pol proposals ike_cert_prop
set security ike policy ike_cert_pol certificate local-certificate lc-igloo-root
set security ike policy ike_cert_pol certificate policy-oids 2.16.840.1.101.3.1.48.2
set security ike policy ike_cert_pol certificate policy-oids 5.16.40.1.101.3.1.55.2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure policy OIDs for certificate validation:

1. Configure the IKE policy:

```
[edit security ike policy ike_cert_pol]
user@host# set mode main
user@host# set proposals ike_cert_prop
user@host# set certificate local-certificate lc-igloo-root
user@host# set certificate policy-oids 2.16.840.1.101.3.1.48.2
user@host# set certificate policy-oids 5.16.40.1.101.3.1.55.2
```

Results

From configuration mode, confirm your configuration by entering the `show security ike policy ike_cert_pol` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show security ike policy ike_cert_pol
mode main;
proposals ike_cert_prop;
certificate {
    local-certificate lc-igloo-root;
    policy-oids [ 2.16.840.1.101.3.1.48.2 5.16.40.1.101.3.1.55.2 ];
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the CA Certificate

Purpose

Display the CA certificate configured on the device.

Action

From operational mode, enter the `show security pki ca-certificate ca-profile ca-tmp` command.

```
user@host> show security pki ca-certificate ca-profile ca-tmp detail
                   Certificate identifier: ca-tmp
                          Certificate version: 3
                          Serial number: 00000047
                          Issuer:
                            Organization: U.S. Government,
                            Organizational unit: DoD, Organizational unit: Testing, Country: US,
                            Common name: Trust Anchor
                          Subject:
                            Organization: U.S. Government,
                            Organizational unit: Dod, Organizational unit: Testing, Country: US,
                            Common name: CA1-PP.01.03
                          Subject string:
                            C=US, O=U.S. Government, OU=Dod, OU=Testing, CN=CA1-PP.01.03
                          Validity:
                            Not before: 01- 1-1998 12:01 UTC
                            Not after: 01- 1-2048 12:01 UTC

?Public key algorithm: rsaEncryption(1024 bits)
  30:81:89:02:81:81:00:cb:fd:78:0c:be:87:ac:cd:c0:33:66:a3:18
  9e:fd:40:b7:9b:bc:dc:66:ff:08:45:f7:7e:fe:8e:d6:32:f8:5b:75
  db:76:f0:4d:21:9a:6e:4f:04:21:4c:7e:08:a1:f9:3d:ac:8b:90:76
  44:7b:c4:e9:9b:93:80:2a:64:83:6e:6a:cd:d8:d4:23:dd:ce:cb:3b
  b5:ea:da:2b:40:8d:ad:a9:4d:97:58:cf:60:af:82:94:30:47:b7:7d
  88:c3:76:c0:97:b4:6a:59:7e:f7:86:5d:d8:1f:af:fb:72:f1:b8:5c
  2a:35:1e:a7:9e:14:51:d4:19:ae:c7:5c:65:ea:f5:02:03:01:00:01
Signature algorithm: sha1WithRSAEncryption
Certificate Policy:
  Policy Identifier = 2.16.840.1.101.3.1.48.2
Use for key: CRL signing, Certificate signing
Fingerprint:
  e0:b3:2f:2e:a1:c5:ee:ad:af:dd:96:85:f6:78:24:c5:89:ed:39:40 (sha1)
  f3:47:6e:55:bc:9d:80:39:5a:40:70:8b:10:0e:93:c5 (md5)
```

Verifying Policy OID Validation

Purpose

If the peer's certificate is successfully validated, IKE and IPsec security associations are established. If the validation of the peer's certificate fails, no IKE security association is established.

Action

From operational mode, enter the `show security ike security-associations` and `show security ipsec security-associations` commands.

```
user@host> show security ike security-associations
node0:
--------------------------------------------------------------------------
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
821765168 UP   88875c981252c1d8  b744ac9c21bde57e  IKEv2          192.0.2.2
1106977837 UP  1a09e32d1e6f20f1  e008278091060acb  IKEv2          198.51.100.202
```

```
user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <213909506 ESP:aes-cbc-192/sha256 8cb9e40a 1295/ unlim - root 500 192.0.2.2
  >213909506 ESP:aes-cbc-192/sha256 8271d2b2 1295/ unlim - root 500 192.0.2.2
  <218365954 ESP:aes-cbc-192/sha256 d0153bc0 1726/ unlim - root 1495 198.51.100.202
  >218365954 ESP:aes-cbc-192/sha256 97611813 1726/ unlim - root 1495 198.51.100.202
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. In this case, check for the PKID_CERT_POLICY_CHECK_FAIL message in the system logs. This message indicates that the peer's certificate chain does not contain a policy OID that is configured on the SRX Series Firewall. Check the **policy-oids** values in the peer's certificate chain with the values configured on the SRX Series Firewall.

It might also be that the peer's certificate chain does not contain any **policy-oids** fields, which are optional fields. If this is the case, certificate validation fails if there are any policy OIDs configured on the SRX Series Firewall.

SEE ALSO

Example: Configuring a Certificate Authority Profile with CRL Locations | 75
Deleting Certificates (CLI Procedure) | 66

Dynamic Updated of Trusted CA Certificates

SUMMARY

Read this topic to understand and configure dynamic update of default trusted CA certificates on your Junos OS devices.

IN THIS SECTION

Understanding Dynamic Update of Trusted CA Certificates

IN THIS SECTION

The Junos OS device like an SRX Series Firewall provides a list of default trusted CA (Certificate Authority) certificates. These certificates are managed dynamically by the Junos OS device. You can also create a custom list of trusted CA certificates and load them into the device. But the custom trusted CA certificates needs to be managed manually. This section focuses on dynamic management of default trusted CA certificates.

With dynamic update of default trusted CA bundle -

- Removal of a CA in the event of compromise is taken care automatically.

- Addition of new CA to the default trusted CA bundle is immediate without having to wait for the new Junos OS release.

To load default trusted CA certificates, see request security pki ca-certificate ca-profile-group load with `filename default` option.

Tasks involved in dynamic update of trusted CA bundle

Following tasks are performed as part of dynamic update of default trusted CA bundle -

- Juniper CDN server (http://signatures.juniper.net/cacert) hosts the default trusted CA certificates.

- The server hosts signed copy of target file and manifest file along with the EE certificate to verify the signed copy of these files. The target file contains a list of default trusted CA certificates (`default-trusted-ca-certs`). The manifest file includes the revision number and date of modification of the default trusted CA bundle.

- Automatic download of trusted CA bundle is enabled by default in Junos OS device. You can either use default or non-default routing instance to connect to the Internet in order to download and update the default trusted CA certificates.

- Public Key Management (PKI) process using PKID securely downloads the default trusted CA bundle (`default-trusted-ca-certs`) from the CDN server into the device.

  NOTE: Dynamic update of trusted CA certificates does not manage any changes to the previously loaded `ca-profile-group`, manually added CA certificates and certificates that are part of other trusted groups.

  See "Configuring Dynamic Updated of Trusted CA Certificates" on page 93.

- Once the `ca-profile-group load` command is issued, PKI process loads the default trusted CA certificates in the background, unblocking the CLI, allowing you to proceed with other tasks.

- If there is no `ca-profile-group` associated with `default-trusted-ca-certs`, with each periodic polling, PKI still downloads the latest copy of trusted CA bundle to the device.

- If a CA certificate is deleted from the default trusted CA list, the PKI process ensures all references to the CA certificate are removed. If any references are present in the `trusted-ca-group`, it only holds the references to `ca-profile` names with actual CA certificates already deleted. See "Configuring Dynamic Updated of Trusted CA Certificates" on page 93.

- PKI process periodically, by default every 24 hours, polls the CDN server for the latest default trusted CA bundle and updates the list for any changes to the trusted CAs in the bundle. If there are any changes, PKI process loads them in the background. You can optionally change the polling duration and also disable this auto-update process. See "Configuring Dynamic Updated of Trusted CA Certificates" on page 93.

Configuring Dynamic Update of Trusted CA Certificates

IN THIS SECTION

Preprequisites

Before configuring dynamic update of default trusted CA certificates, ensure to meet the following prerequisites -

- Basic configuration of Junos OS device is completed.
- Your Junos OS device is reachable to Juniper CDN server. You can use non-default routing instance as well to connect to Internet to download the default trusted CA certificates. Ensure non-default routing instance is configured prior to configuring dynamic update of trusted CA certificates. Contact Juniper sales for Juniper CDN server details.
- For custom CDN server, ensure to have latest CA certificates and the URL. Configuration of custom CDN server is out of scope of this topic.

Based on your requirements, navigate to the following tasks to configure dynamic update of default trusted CA bundle.

Checking connectivity to the CDN server

IN THIS SECTION

Overview

Use the following CLI to check connectivity to the CDN server for downloading default trusted CA certificates. This command downloads the manifest file and displays the trusted-ca-bundle version available in the CDN server.

See request security pki ca-certificate ca-profile-group default-trusted-ca-certs, for details about the command.

Configuration

1. To check connectivity to the CDN server from operational mode of the Junos OS device -

```
user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
download check-server
```

Enabling automatic download of default trusted CA certificates

IN THIS SECTION

Overview

Juniper Networks regularly updates the default trusted CA certificates on Juniper CDN server and makes it available for download on Junos OS device. Automatic download of default trusted CA certificates is enabled by default on Junos OS device. You can customize the configuration and load the latest default trusted CA certificates at specified intervals. The default periodicity is 24 hours when you don’t specify a value. When you use the default Juniper CDN Server (http://signatures.juniper.net/cacert), no separate configuration is needed.

This example shows how to enable automatic download of default trusted CA certificates on Junos OS device using default configuration settings. See default-trusted-ca-certs (Security) for details about the configuration statement. Loading of the downloaded default trusted CA certificates automatically happens in the background using the statement request security pki ca-certificate ca-profile-group load command. You don't have to explicitly run this command to load the certificates.

Configuration

As automatic download of default trusted CA certificates is enabled by default, no separate configuration is needed.

Providing custom configuration for automatic download of default trusted CA certificates

IN THIS SECTION

Overview

In this example, you provide following custom configuration while enabling the automatic download of custom CA certificates -

- Configure the Junos OS device to download and install the default trusted CA certificates every 48 hours.
- Specify the custom CDN server reachable via the URL signatures.example.net.
- Specify non-default routing instance to reach the CDN server.

See default-trusted-ca-certs (Security) for details about the configuration statement.

Configuration

Configuration

1. Set the periodicity of download and load operations to 48 hours. This CLI automatically loads the certificates into the Junos OS device.

```
[edit]
user@host# set security pki default-trusted-ca-certs automatic-download interval hours 48
```

2. Specify the custom URL.

```
[edit]
user@host# set security pki default-trusted-ca-certs automatic-download url
signatures.example.net
```

3. Specify the routing instance.

```
[edit]
user@host# set security pki default-trusted-ca-certs automatic-download routing-instance RI1
```

4. Commit the configuration.

```
[edit]
user@host# commit
```

Downloading default trusted CA certificates explicitly

IN THIS SECTION

Overview

Use the following CLI to manually download default trusted CA certificates to the Junos OS device from the CDN server. This command is in addition to automatic download of default trusted CA certs at regular intervals.

See request security pki ca-certificate ca-profile-group default-trusted-ca-certs for details about the command.

Configuration

Configuration

1. To explicitly download default trusted CA certificates from operational mode of the Junos OS device -

```
user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
download
```

Checking the download status of default trusted CA certificates

IN THIS SECTION

Overview

Use the following CLI to check the download status of default trusted CA certificates on the Junos OS device from the CDN server. This command displays the version number and version date. You can use this command to check the previous downloaded version and date.

See request security pki ca-certificate ca-profile-group default-trusted-ca-certs for details about the command.

Configuration

Configuration

1. To check the version number and version date available on the Junos OS device -

```
user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
download status
```

Deactivating automatic download of trusted CA certificates

IN THIS SECTION

Overview

Automatic download is enabled by default. This example shows how to deactivate the automatic download of default trusted CA certificates, though we don't recommend.

See default-trusted-ca-certs (Security) for details about the configuration statement.

Configuration

Configuration

1. To deactivate automatic download of default trusted CA certificates -

```
[edit]
user@host# set security pki default-trusted-ca-certs automatic-download deactivate
```

2. Commit the configuration.

```
[edit]
user@host# commit
```

RELATED DOCUMENTATION

request security pki ca-certificate ca-profile-group load

default-trusted-ca-certs (Security)

request security pki ca-certificate ca-profile-group default-trusted-ca-certs

ACME Protocol

IN THIS SECTION

Understanding ACME Protocol

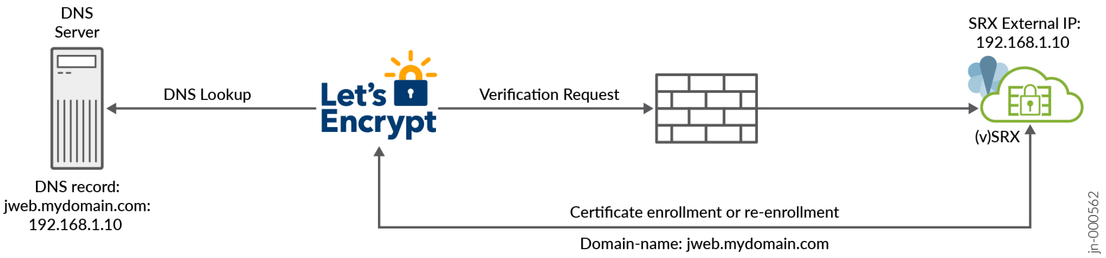
IN THIS SECTION

Automated Certificate Management Environment (ACME) protocol is a new PKI enrollment standard used by several PKI servers such as Let’s Encrypt. The Let’s encrypt certificate allows for free usage of Web server certificates in SRX Series Firewalls, and this can be used in Juniper Secure Connect and J-Web. The Junos OS automatically re-enroll Let’s Encrypt certificates on occurance of every 25 days.

The ACME protocol allows the enrollment of certificates from Let’s Encrypt server or ACME enabled servers. The SRX Series Firewalls enrolls the certificates from Let’s Encrypt server and Juniper Secure Connect validates the certificates without copying and downloading any CA certificates.

When using Let’s Encrypt, ensure that the Let’s Encrypt server is able to resolve the domain name to the IP address of the SRX Series Firewall interface as shown in Figure 10 on page 100. It must be able to reach the SRX Series Firewall interface on TCP port 80. During the certificate enrollment, the SRX Series Firewall will temporarily allow this incoming request automatically. If your SRX Series Firewall or an intermediate firewall or a router is blocking the TCP port 80, certificate enrollment will fail.

Figure 10: Name Resolution for Let's Encrypt



Limitations

- ACME specification - The dns-01 and external account binding are not supported.
- ACME cannot be used when J-Web listen to port 80
- Wildcard certificate is not supported such as `*.mydomain.com`, instead you can enroll multiple dns names.

Enroll Local Certificate Using Let's Encrypt Server

This example shows how to enroll the local certificate using Let's Encrypt.

1. Specify the CA profile.

```
[edit]

user@host#
set security pki ca-profile ISRG_Root_X1 ca-identity ISRG_Root_X1

user@host# set security pki ca-profile ISRG_Root_X1 revocation-check disable
user@host# set security pki ca-profile Lets_Encrypt ca-identity Lets_Encrypt

user@host#
set security pki ca-profile Lets_Encrypt enrollment url https://acme-v02.api.letsencrypt.org/
directory
```

2. Commit the configuration.

```
[edit]
user@host# commit
```

3. Load the CA certificate.

```
[edit]
user@host> request security pki ca-certificate load ca-profile ISRG_Root_X1 filename
ISRG_Root_X1.pem
```

4. Create ACME key ID.

```
[edit]
user@host> request security pki generate-key-pair size 2048 type rsa acme-key-id mydomain
```

5. Preparing enrollment of local certificate.

```
[edit]
user@host> request security pki generate-key-pair size 2048 type rsa certificate-id service-
mydomain
```

6. Enroll a certificate with one domain name.

```
[edit]
user@host> request security pki local-certificate enroll acme acme-key-id mydoamin
certificate-id service-mydomain ca-profile Lets_Encrypt domain-name jweb.mydomain.com email
jweb@acmejnpr.net letsencrypt-enrollment yes terms-of-service agree
```

Enroll a certificate with multiple domain names.

```
[edit]
user@host> request security pki local-certificate enroll acme acme-key-id mydomain
certificate-id service-mydomain ca-profile Lets_Encrypt domain-name jweb.mydomain.com,remote-
acess.mydomain.com  email jweb@acmejnpr.net letsencrypt-enrollment yes terms-of-service agree
```

7. Once the enrollment is finished the issued certificate will be loaded in certificate-id service-mydomian.

Manual Re-Enroll Local Certificate

To re-enroll a local certificate online:

1. Initiate the re-enrollment request.

   ```
   [edit]
   user@host> request security pki local-certificate re-enroll acme acme-key-id mydomain
   certificate-id serice-mydomain ca-profile Lets_Encrypt re-generate-keypair
   ```

2. Once the re-enrollment is finished the issued certificate will be loaded in certificate-id service-mydomian.

Delete ACME Account

To delete the ACME account:

1. Delete the ACME account.

   ```
   [edit]
   user@host> clear security pki acme account acme-key-id mydomain ca-profile Lets_Encrypt
   ```

   You can delete the ACME account key only if the ACME is activated or created by the enrollment.

Configure Multiple Certificate Types to Establish IKE and IPsec SA

IN THIS SECTION

This example shows how to configure multiple certificate types to establish IKE and IPsec SA.

Starting in Junos OS Release 22.4R1, you can establish tunnels irrespective of the certificate type used on the initiator and responder if authentication-method is configured as `certificates` in IKE proposal using the `set security ike proposal` *`ike_proposal_name`* `authentication-method certificates` command.

You can view the certificate enrolled using `show security pki local-certificate certificate-id` *`certificate-name`* `detail` command.

You can verify the enrolled certificate using the `request security pki local-certificate verify certificate-id` *`certificate-name`* command.

Requirements

Before you begin:

- Ensure that you have certificates enrolled on your devices, see Certificate Enrollment.

  You can verify the certificates enrolled on your devices using the `request security pki local-certificate certificate-id` *`certificate-name`* `detail` command.

- Ensure that you have IKE package installed, to verify the installed IKE package use the `show version | match ike` operational command.

If you don't have the IKE package installed on the device, you can install the IKE package using the operational command `request system software add optional://junos-ike.tgz`, for more information, see Enabling IPsec VPN Feature Set.

Overview

This example configures multiple certificate types to establish IKE and IPsec SA between on SRX_A and on SRX_B.

NOTE: In this example, we have enrolled the RSA certificate on SRX_A and the ECDSA certificate on SRX_B devices. For more information about how to install the certificates, see Certificate Enrollment.

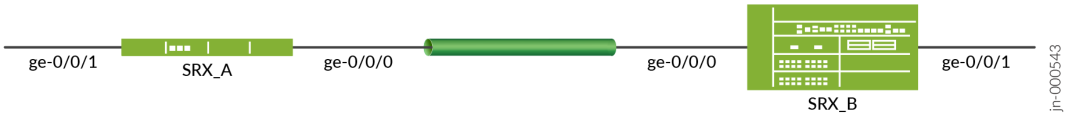
Table 4: Topology Setup for SRX_A and SRX_B Devices

Device Name	Interface Used	IKE Gateway Address	IKE Gateway Local IP Address
SRX_A	ge-0/0/0	192.168.1.2	192.168.1.1
SRX_B	ge-0/0/0	192.168.1.1	192.168.1.2

Topology

The Figure 11 on page 104 describes topology for multiple certificate types support configuration.

Figure 11: Multiple Certificate Types Support Configuration Example



Configuration

IN THIS SECTION

Configuring SRX_A

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24
set interfaces st0 unit 1 family inet
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/1
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0
set security zones security-zone VPN interfaces st0.1
set security policies from-zone VPN to-zone trust policy 1 match source-address any
set security policies from-zone VPN to-zone trust policy 1 match destination-address any
set security policies from-zone VPN to-zone trust policy 1 match application any
set security policies from-zone VPN to-zone trust policy 1 then permit
set security policies from-zone trust to-zone VPN policy 1 match source-address any
set security policies from-zone trust to-zone VPN policy 1 match destination-address any
set security policies from-zone trust to-zone VPN policy 1 match application any
set security policies from-zone trust to-zone VPN policy 1 then permit
set security policies default-policy deny-all
set security ike proposal IKE_PROP authentication-method certificates
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha-256
set security ike proposal IKE_PROP encryption-algorithm aes-128-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate r0_rsa_crt
set security ike gateway IKE_GW ike-policy IKE_POL
set security ike gateway IKE_GW address 192.168.1.2
set security ike gateway IKE_GW external-interface ge-0/0/0
set security ike gateway IKE_GW local-address 192.168.1.1
set security ike gateway IKE_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-192-cbc
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN bind-interface st0.1
set security ipsec vpn IPSEC_VPN ike gateway IKE_GW
set security ipsec vpn IPSEC_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN establish-tunnels on-traffic
```

Step-by-step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI Configuration Mode Overvie in the CLI User Guide.

To configure multiple certificate types to establish IKE and IPsec SA:

1. View the certificates enrolled on your devices using the `show security pki local-certificate certificate-id` *`certificate-name`* `detail` command.

   Install the certificate on your device if your device does not have the certificates enrolled. For more information, see Certificate Enrollment.

2. Configure interfaces.

   ```
   user@srxa# set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/24
   user@srxa# set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24
   user@srxa# set interfaces st0 unit 1 family inet
   ```

3. Configure security zones and the security policy.

   ```

   user@srxa# set security zones security-zone trust host-inbound-traffic  system-services all
   user@srxa# set security zones security-zone trust host-inbound-traffic  protocols all
   user@srxa# set security zones security-zone trust interfaces ge-0/0/1
   user@srxa# set security zones security-zone untrust host-inbound-traffic  system-services ike
   ```

```
user@srxa# set security zones security-zone untrust interfaces ge-0/0/0
user@srxa# set security zones security-zone VPN interfaces st0.1
user@srxa# set security policies from-zone VPN to-zone trust policy 1 match source-address
any
user@srxa# set security policies from-zone VPN to-zone trust policy 1 match destination-
address any
user@srxa# set security policies from-zone VPN to-zone trust policy 1 match application any
user@srxa# set security policies from-zone VPN to-zone trust policy 1 then permit
user@srxa# set security policies from-zone trust to-zone VPN policy 1 match source-address
any
user@srxa# set security policies from-zone trust to-zone VPN policy 1 match destination-
address any
user@srxa# set security policies from-zone trust to-zone VPN policy 1 match application any
user@srxa# set security policies from-zone trust to-zone VPN policy 1 then permit
user@srxa# set security policies default-policy deny-all
```

4. Configure the IKE proposal.

```
[edit]
user@srxa# set security ike proposal IKE_PROP authentication-method certificates
user@srxa# set security ike proposal IKE_PROP dh-group group5
user@srxa# set security ike proposal IKE_PROP authentication-algorithm sha-256
user@srxa# set security ike proposal IKE_PROP encryption-algorithm aes-128-cbc
```

5. Configure the IKE policy.

```
[edit]
user@srxa# set security ike policy IKE_POL proposals IKE_PROP
user@srxa# set security ike policy IKE_POL certificate local-certificate r0_rsa_crt
```

6. Configure the IKE gateway.

```
[edit]
user@srxa# set security ike gateway IKE_GW ike-policy IKE_POL
user@srxa# set security ike gateway IKE_GW address 192.168.1.2
user@srxa# set security ike gateway IKE_GW external-interface ge-0/0/0
user@srxa# set security ike gateway IKE_GW local-address 192.168.1.1
user@srxa# set security ike gateway IKE_GW version v2-only
```

7. Configure the IPsec proposal.

```
[edit]
user@srxa# set security ipsec proposal IPSEC_PROP protocol esp
user@srxa# set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@srxa# set security ipsec proposal IPSEC_PROP encryption-algorithm aes-192-cbc
```

8. Configure the IPsec policy.

```
[edit]
user@srxa# set security ipsec policy IPSEC_POL proposals IPSEC_PROP
```

9. Configure the IPsec VPN.

```
[edit]
user@srxa# set security ipsec vpn IPSEC_VPN bind-interface st0.1
user@srxa# set security ipsec vpn IPSEC_VPN ike gateway IKE_GW
user@srxa# set security ipsec vpn IPSEC_VPN ike ipsec-policy IPSEC_POL
user@srxa# set security ipsec vpn IPSEC_VPN establish-tunnels on-traffic
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike` and, `show security ipsec` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@srxa# show interfaces
ge-0/0/0 {
    description untrust;
    unit 0 {
        family inet {
            address 192.168.1.1/24;
            }
        }
    }
ge-0/0/1 {
    description trust;
    unit 0 {
        family inet {
            address 172.16.1.1/24;
            }
        }
    }
st0 {
    unit 1 {
        family inet;
    }
}

[edit]
user@srxa# show security ike
proposal IKE_PROP {
    authentication-method certificates;
    dh-group group5;
    authentication-algorithm sha-256;
    encryption-algorithm aes-128-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    certificate {
        local-certificate r0_crt_rsa;
    }
}
gateway IKE_GW {
    ike-policy IKE_POL;
    address 192.168.1.2;
    external-interface ge-0/0/0;
    local-address 192.168.1.1;
    version v2-only;
}

[edit]
user@srxa# show security ipsec
    proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-192-cbc;
}
policy IPSEC_POL {
    proposals IPSEC_PROP;
```

```
}
vpn IPSEC_VPN {
    bind-interface st0.1;
    ike {
        gateway IKE_GW;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels on-traffic;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring SRX_B

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.2/24
set interfaces ge-0/0/1 unit 0 family inet address 172.18.1.2/24
set interfaces st0 unit 1 family inet
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/1
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0
set security zones security-zone VPN interfaces st0.1
set security policies from-zone VPN to-zone trust policy 1 match source-address any
set security policies from-zone VPN to-zone trust policy 1 match destination-address any
set security policies from-zone VPN to-zone trust policy 1 match application any
set security policies from-zone VPN to-zone trust policy 1 then permit
set security policies from-zone trust to-zone VPN policy 1 match source-address any
set security policies from-zone trust to-zone VPN policy 1 match destination-address any
set security policies from-zone trust to-zone VPN policy 1 match application any
set security policies from-zone trust to-zone VPN policy 1 then permit
set security policies default-policy deny-all
set security ike proposal IKE_PROP authentication-method certificates
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha-256
set security ike proposal IKE_PROP encryption-algorithm aes-128-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate r1_crt_ecdsa384
set security ike gateway IKE_GW ike-policy IKE_POL
set security ike gateway IKE_GW address 192.168.1.1
set security ike gateway IKE_GW external-interface ge-0/0/0
set security ike gateway IKE_GW local-address 192.168.1.2
set security ike gateway IKE_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-192-cbc
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN bind-interface st0.1
set security ipsec vpn IPSEC_VPN ike gateway IKE_GW
set security ipsec vpn IPSEC_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN establish-tunnels on-traffic
```

Step-by-step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI Configuration Mode Overview in the CLI User Guide.

To configure multiple certificate types to establish IKE and IPsec SA:

1. View the certificates enrolled on your devices using the `request security pki local-certificate certificate-id` *`certificate-name`* `detail` command.

   Install the certificate on your device if your device does not have the certificates enrolled. For more information, see Certificate Enrollment.

2. Configure interfaces.

   ```
   user@srxb# set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.2/24
   user@srxb# set interfaces ge-0/0/1 unit 0 family inet address 172.18.1.2/24
   user@srxb# set interfaces st0 unit 1 family inet
   ```

3. Configure security zones and the security policy.

   ```

   user@srxb# set security zones security-zone trust host-inbound-traffic  system-services all
   user@srxb# set security zones security-zone trust host-inbound-traffic  protocols all
   user@srxb# set security zones security-zone trust interfaces ge-0/0/1
   ```

```
user@srxb# set security zones security-zone untrust host-inbound-traffic  system-services ike
user@srxb# set security zones security-zone untrust interfaces ge-0/0/0
user@srxb# set security zones security-zone VPN interfaces st0.1
user@srxb# set security policies from-zone VPN to-zone trust policy 1 match source-address
any
user@srxb# set security policies from-zone VPN to-zone trust policy 1 match destination-
address any
user@srxb# set security policies from-zone VPN to-zone trust policy 1 match application any
user@srxb# set security policies from-zone VPN to-zone trust policy 1 then permit
user@srxb# set security policies from-zone trust to-zone VPN policy 1 match source-address
any
user@srxb# set security policies from-zone trust to-zone VPN policy 1 match destination-
address any
user@srxb# set security policies from-zone trust to-zone VPN policy 1 match application any
user@srxb# set security policies from-zone trust to-zone VPN policy 1 then permit
user@srxb# set security policies default-policy deny-all
```

4. Configure the IKE proposal.

```
[edit]
user@srxb# set security ike proposal IKE_PROP authentication-method certificates
user@srxb# set security ike proposal IKE_PROP dh-group group5
user@srxb# set security ike proposal IKE_PROP authentication-algorithm sha-256
user@srxb# set security ike proposal IKE_PROP encryption-algorithm aes-128-cbc
```

5. Configure the IKE policy.

```
[edit]
user@srxb# set security ike policy IKE_POL proposals IKE_PROP
user@srxb# set security ike policy IKE_POL certificate local-certificate r1_crt_ecdsa384
```

6. Configure the IKE gateway.

```
[edit]
user@srxb# set security ike gateway IKE_GW ike-policy IKE_POL
user@srxb# set security ike gateway IKE_GW address 192.168.1.1
user@srxb# set security ike gateway IKE_GW external-interface ge-0/0/0
user@srxb# set security ike gateway IKE_GW local-address 192.168.1.2
user@srxb# set security ike gateway IKE_GW version v2-only
```

7. Configure the IPsec proposal.

   ```
   [edit]
   user@srxb# set security ipsec proposal IPSEC_PROP protocol esp
   user@srxb# set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
   user@srxb# set security ipsec proposal IPSEC_PROP encryption-algorithm aes-192-cbc
   ```

8. Configure the IPsec policy.

   ```
   [edit]
   user@srxb# set security ipsec policy IPSEC_POL proposals IPSEC_PROP
   ```

9. Configure the IPsec VPN.

   ```
   [edit]
   user@srxb# set security ipsec vpn IPSEC_VPN bind-interface st0.1
   user@srxb# set security ipsec vpn IPSEC_VPN ike gateway IKE_GW
   user@srxb# set security ipsec vpn IPSEC_VPN ike ipsec-policy IPSEC_POL
   user@srxb# set security ipsec vpn IPSEC_VPN establish-tunnels immediately
   ```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike` and, `show security ipsec` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@srxb# show interfaces
ge-0/0/0 {
    description untrust;
    unit 0 {
        family inet {
            address 192.168.1.2/24;
            }
        }
    }
ge-0/0/1 {
    description trust;
```

```
    unit 0 {
        family inet {
            address 172.18.1.2/24;
            }
        }
    }
st0 {
    unit 1 {
        family inet;
    }
}

[edit]
user@srxb# show security ike
proposal IKE_PROP {
    authentication-method certificates;
    dh-group group5;
    authentication-algorithm sha-256;
    encryption-algorithm aes-128-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    certificate {
        local-certificate r1_crt_ecdsa384;
    }
}
gateway IKE_GW {
    ike-policy IKE_POL;
    address 192.168.1.1;
    external-interface ge-0/0/0;
    local-address 192.168.1.2;
    version v2-only;
}

[edit]
user@srxb# show security ipsec
    proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-192-cbc;
}
policy IPSEC_POL {
    proposals IPSEC_PROP;
```

```
}
vpn IPSEC_VPN {
    bind-interface st0.1;
    ike {
        gateway IKE_GW;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verify SRX_A

The sample outputs shown are on SRX-A.

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ike security-associations` command.

```
user@srxa> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
32      UP     6723643250f0f357  f6295f11b0d7c8ab  IKEv2          192.168.1.2
```

From operational mode, enter the `show security ipsec security-associations` command.

```
user@srxa> show security ipsec security-associations
  Total active tunnels: 1     Total IPsec sas: 1
  ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <500033 ESP:aes-cbc-192/sha256 0x5f156c1b 2750/ unlim - root 500 192.168.1.2
  >500033 ESP:aes-cbc-192/sha256 0x7ea065e7 2750/ unlim - root 500 192.168.1.2
```

From operational mode, enter the `show security ike security-associations detail` command.

```
user@srxa> show security ike security-associations detail
  IKE peer 192.168.1.2, Index 32, Gateway Name: IKE_GW
  Role: Responder, State: UP
  Initiator cookie: 6723643250f0f357, Responder cookie: f6295f11b0d7c8ab
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local gateway interface: ge-0/0/0.0
  Routing instance: default
  Local: 192.168.1.1:500, Remote: 192.168.1.2:500
  Lifetime: Expires in 28165 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: 192.168.1.2
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes128-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                1346
   Output bytes  :                1887
   Input  packets:                    3
   Output packets:                    4
   Input  fragmented packets:       2
   Output fragmented packets:       3
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500033

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 192.168.1.1:500, Remote: 192.168.1.2:500
    Local identity: 192.168.1.1
    Remote identity: 192.168.1.2
    Flags: IKE SA is created

  IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                  Responder stats:
    Request Out             : 0                       Request In             :
0
    Response In             : 0                       Response Out           :
0
    No Proposal Chosen In   : 0                       No Proposal Chosen Out :
0
    Invalid KE In           : 0                       Invalid KE Out         :
0
    TS Unacceptable In      : 0                       TS Unacceptable Out    :
0
    Res DH Compute Key Fail : 0                       Res DH Compute Key Fail:
0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0
```

From operational mode, enter the `show security ipsec security-associations detail` command.

```
user@srxa> show security ipsec security-associations detail
  ID: 500033 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
  Local Identity: ipv4(0.0.0.0-255.255.255.255)
  Remote Identity: ipv4(0.0.0.0-255.255.255.255)
  TS Type: proxy-id
  Version: IKEv2
  PFS group: N/A
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Tunnel MTU: 0, Policy-name:
IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Tunnel events:
    Thu Mar 09 2023 22:41:36: IPsec SA negotiation succeeds (1 times)
  Location: FPC 0, PIC 0, KMD-Instance 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0x5f156c1b, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2895 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2286 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-on-traffic
    IKE SA Index: 32
  Direction: outbound, SPI: 0x7ea065e7, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2895 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2286 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-on-traffic
    IKE SA Index: 32
```

From operational mode, enter the `show security pki local-certificate certificate-id r0_rsa_cr detail` command.

```
user@srxa> show security pki local-certificate certificate-id r0_rsa_crt detail
  LSYS: root-logical-system
Certificate identifier: r0_rsa_crt
  Certificate version: 3

  Serial number:
    hexadecimal: 0x0186a62478ae8f0cdd766eb38dbd53
    decimal: 79233029077573018470071062263063 87
```

```
    decimal: 7923302907757301847007106226306387
  Issuer:
    Organization: juniper, Country: India, Common name: Root-CA
  Subject:
    Organization: juniper, Organizational unit: marketing, State: california, Locality:
sunnyvale, Common name: r0, Domain component: juniper
  Subject string:
    DC=juniper, CN=r0, OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
  Alternate subject: "r0@juniper.net", r0.juniper.net, 192.168.1.1
  Cert-Chain: Root-CA
  Validity:
    Not before: 03- 3-2023 05:54 UTC
    Not after: 06- 6-2027 12:36 UTC
  Public key algorithm: rsaEncryption(2048 bits)
    30:82:01:0a:02:82:01:01:00:b0:e5:53:8d:7e:20:fa:6b:21:c2:d1
    2b:48:8f:af:c3:eb:8b:23:4a:f7:c5:1f:cf:2c:6a:b3:2e:8a:ef:1b
    f7:97:aa:fd:1d:ab:1c:76:9b:40:a3:ac:bb:49:f6:93:f9:e1:4e:62
    df:3d:ca:e5:d2:95:9c:a0:f4:2b:d7:7e:1d:20:94:69:a8:e4:cf:dc
    15:90:4c:be:1d:d8:1c:52:08:3a:d1:05:a3:bb:2f:8f:31:0c:6b:21
    ef:76:c3:c7:fb:be:4a:cb:da:cc:8d:04:3a:75:0c:eb:5d:e2:f6:13
    50:fe:39:67:c0:77:2f:32:b0:5e:38:6f:9c:79:b3:5d:f3:57:f4:f8
    42:f5:22:5b:6c:58:67:90:4e:1e:ec:6a:03:e2:c0:87:65:02:ca:da
    6f:95:0a:8c:2a:fd:45:4f:3a:b5:ef:18:05:1c:54:e6:fe:45:bb:73
    53:81:b2:c6:b7:36:36:57:6d:9c:d3:d9:80:e7:d6:85:92:74:32:88
    16:01:03:27:57:76:8e:5e:d6:73:ac:bf:68:fd:6d:a1:2a:8f:f5:3a
    29:b0:c9:44:9b:c8:46:c1:bf:c0:52:2a:f0:51:be:b5:f6:e1:f5:3e
    96:1d:3a:42:29:28:d3:cf:60:b9:eb:24:04:47:d3:f1:3f:5e:38:fc
    7f:33:f6:94:9d:02:03:01:00:01
  Signature algorithm: sha256WithRSAEncryption
  Fingerprint:
    4d:f6:89:c5:d6:3c:74:73:db:3e:f6:4b:1e:26:6c:c1:1c:1d:a7:4d (sha1)
    6b:1c:a8:1f:de:5a:9b:3e:d5:c4:85:29:af:3f:82:f2 (md5)

6b:7a:b5:d1:57:cf:75:9d:1f:63:b9:f6:49:e4:4e:b3:13:2c:83:f1:f7:25:44:6f:45:2f:0d:2f:ae:a8:80:85
(sha256)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

From operational mode, enter the `show security pki ca-certificate ca-profile Root-CA detail` command.

```
user@srxa> show security pki ca-certificate ca-profile Root-CA detail
  LSYS: root-logical-system
  CA profile: Root-CA
Certificate identifier: Root-CA
  Certificate version: 3

  Serial number:
    hexadecimal: 0x00000440
    decimal: 1088
  Issuer:
    Organization: juniper, Country: India, Common name: Root-CA
  Subject:
    Organization: juniper, Country: India, Common name: Root-CA
  Subject string:
    C=India, O=juniper, CN=Root-CA
  Validity:
    Not before: 06- 7-2022 12:36 UTC
    Not after: 06- 6-2027 12:36 UTC
  Public key algorithm: rsaEncryption(2048 bits)
    30:82:01:0a:02:82:01:01:00:cd:9c:e6:9f:62:6c:49:15:c2:da:eb
    8e:e6:e5:a1:88:40:d8:b5:2e:5b:1a:0e:de:96:d7:0b:19:f9:03:44
    98:49:d5:cc:a8:90:2b:7f:1b:58:7b:1f:26:92:18:4c:2d:37:65:5c
    9f:0f:6e:10:b5:34:6f:2d:b5:9c:27:3b:a6:b1:b5:a0:e2:a6:92:3d
    e4:68:fe:5d:71:06:6f:ce:e6:0f:0f:e3:94:2a:23:57:98:a0:6a:9c
    e0:52:a2:47:ff:ce:b0:47:bd:36:95:80:a7:af:d2:49:b1:5d:2a:3d
    28:e4:95:06:b8:b3:d9:07:11:3c:13:af:c6:e2:51:08:22:82:2d:ec
    4f:26:40:b0:b0:55:2d:6e:c0:c8:19:34:a7:99:5a:bc:58:98:69:ae
    04:d6:6d:ec:4a:c9:55:a5:ff:00:cb:3b:02:85:fa:02:a1:5c:c1:9d
    6d:44:b8:95:8f:77:c0:53:fc:7f:a4:09:a3:25:1c:4a:e2:9d:0c:81
    08:b4:c8:b8:0d:bc:94:75:54:75:57:4f:d3:a4:17:0d:5d:1a:f3:c1
    1d:5d:73:2f:fe:8b:cb:fc:1f:93:87:72:d6:be:df:86:d7:e6:d1:c7
    0d:00:1a:6e:58:db:6a:1c:2f:1d:17:46:9a:f2:69:b4:21:db:08:5d
    8d:ab:30:7d:7f:02:03:01:00:01
  Signature algorithm: sha256WithRSAEncryption
  Distribution CRL:
     http://10.102.40.55:8080/crl-as-der/currentcrl-11.crl?id=11
  Use for key: CRL signing, Certificate signing, Key encipherment, Digital signature
  Fingerprint:
    8b:84:60:2a:58:5b:80:f0:b9:ae:25:9f:67:3d:d6:81:ee:43:6c:d4 (sha1)
    ab:ec:4d:fe:d4:04:9c:c9:79:1d:9a:33:4e:6d:78:f6 (md5)
```

```
9d:f0:c0:a0:93:74:11:53:d3:4d:2d:75:d3:60:37:5f:fb:b7:a9:67:42:cd:7c:3c:0e:0f:9b:58:36:3c:14:f5
(sha256)
```

Verify SRX_B

The sample outputs shown are on SRX-B.

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ike security-associations` command.

```
user@srxb> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
56042   UP     6723643250f0f357  f6295f11b0d7c8ab  IKEv2          192.168.1.1
```

From operational mode, enter the `show security ipsec security-associations` command.

```
user@srxb> show security ipsec security-associations
Total active tunnels: 1     Total IPsec sas: 1
  ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <500230 ESP:aes-cbc-192/sha256 0x7ea065e7 2638/ unlim - root 500 192.168.1.1
  >500230 ESP:aes-cbc-192/sha256 0x5f156c1b 2638/ unlim - root 500 192.168.1.1
```

From operational mode, enter the `show security ike security-associations detail` command.

```
user@srxb> show security ike security-associations detail
  IKE peer 192.168.1.1, Index 56042, Gateway Name: IKE_GW
  Role: Responder, State: UP
  Initiator cookie: 6723643250f0f357, Responder cookie: f6295f11b0d7c8ab
  Exchange type: IKEv2, Authentication method: ECDSA-384-signatures
  Local gateway interface: ge-0/0/0.0
  Routing instance: default
  Local: 192.168.1.2:500, Remote: 192.168.1.1:500
  Lifetime: Expires in 18995 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: 192.168.1.1
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes128-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2934
   Output bytes  :                 2379
   Input  packets:                   10
   Output packets:                    9
   Input  fragmented packets:       3
   Output fragmented packets:       2
  IPSec security associations: 8 created, 3 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500230

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 192.168.1.2:500, Remote: 192.168.1.1:500
    Local identity: 192.168.1.2
    Remote identity: 192.168.1.1
    Flags: IKE SA is created

  IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                  Responder stats:
    Request Out             : 1                       Request In              :
2
    Response In             : 1                       Response Out            :
2
    No Proposal Chosen In   : 0                       No Proposal Chosen Out :
0
    Invalid KE In           : 0                       Invalid KE Out          :
0
    TS Unacceptable In      : 0                       TS Unacceptable Out     :
0
    Res DH Compute Key Fail : 0                       Res DH Compute Key Fail:
0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0
```

From operational mode, enter the `show security ipsec security-associations detail` command.

```
user@srxb> show security ipsec security-associations detail
  ID: 500230 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 192.168.1.2, Remote Gateway: 192.168.1.1
  Local Identity: ipv4(0.0.0.0-255.255.255.255)
  Remote Identity: ipv4(0.0.0.0-255.255.255.255)
  TS Type: proxy-id
  Version: IKEv2
  PFS group: N/A
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Tunnel MTU: 0, Policy-name: 
IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Tunnel events:
    Thu Mar 02 2023 22:26:16: IPsec SA negotiation succeeds (1 times)
  Location: FPC 0, PIC 0, KMD-Instance 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0x7ea065e7, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2633 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2002 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-on-traffic
    IKE SA Index: 56042
  Direction: outbound, SPI: 0x5f156c1b, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2633 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2002 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-on-traffic
    IKE SA Index: 56042
```

From operational mode, enter the `show security pki local-certificate certificate-id r1_crt_ecdsa384 detail` command.

```
user@srxb> show security pki local-certificate certificate-id r1_crt_ecdsa384 detail
  LSYS: root-logical-system
Certificate identifier: r1_crt_ecdsa384
  Certificate version: 3

  Serial number:
    hexadecimal: 0x0186a6254347a38063946d08595a55
    decimal: 7923303152683216740296668848151125
  Issuer:
    Organization: juniper, Country: India, Common name: root-ecdsa-384
  Subject:
    Organization: juniper, Organizational unit: marketing, State: california, Locality:
sunnyvale, Common name: r1_spk1, Domain component: juniper
  Subject string:
    DC=juniper, CN=r1_spk1, OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
  Alternate subject: "r1_spk1@juniper.net", r1_spk1.juniper.net, 192.168.2
  Cert-Chain: root-ecdsa-384
  Validity:
    Not before: 03- 3-2023 05:55 UTC
    Not after: 06- 6-2027 13:21 UTC
  Public key algorithm: ecdsaEncryption(384 bits)
    04:c2:ba:19:dc:0d:62:a7:94:7b:9b:1d:4d:ff:a1:e1:44:b5:57:a7
    cb:7d:33:6b:35:87:b8:e4:ca:44:b1:6c:6d:63:ae:6f:3c:31:7c:7e
    65:99:b3:2d:a3:76:30:23:e5:0e:34:e1:28:54:d6:3e:d3:8b:de:b6
    b9:45:05:82:6f:1d:20:b7:6f:3c:ce:a2:13:a2:b4:37:0b:db:35:1e
    20:54:b5:06:9d:f8:7f:19:7b:c5:d7:7b:57:8b:28:31:d3
  Signature algorithm: ecdsa-with-SHA384
  Fingerprint:
    9b:cb:5a:57:a8:60:a0:ee:5c:be:59:4c:db:35:39:d3:b7:29:ef:b1 (sha1)
    ef:b5:e3:be:35:1b:6e:02:0b:61:11:a5:53:07:b4:89 (md5)

8f:86:d0:12:ea:bc:a8:81:a8:17:3a:f9:03:e4:91:57:20:9c:11:bc:a4:dd:d1:7f:d1:48:3f:5b:d9:fb:93:32
(sha256)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

s

From operational mode, enter the `show security pki ca-certificate ca-profile Root-CA detail` command.

```
user@srxb> show security pki ca-certificate ca-profile Root-CA detail
  LSYS: root-logical-system
  CA profile: Root-CA
Certificate identifier: Root-CA
  Certificate version: 3

  Serial number:
    hexadecimal: 0x00000440
    decimal: 1088
  Issuer:
    Organization: juniper, Country: India, Common name: Root-CA
  Subject:
    Organization: juniper, Country: India, Common name: Root-CA
  Subject string:
    C=India, O=juniper, CN=Root-CA
  Validity:
    Not before: 06- 7-2022 12:36 UTC
    Not after: 06- 6-2027 12:36 UTC
  Public key algorithm: rsaEncryption(2048 bits)
    30:82:01:0a:02:82:01:01:00:cd:9c:e6:9f:62:6c:49:15:c2:da:eb
    8e:e6:e5:a1:88:40:d8:b5:2e:5b:1a:0e:de:96:d7:0b:19:f9:03:44
    98:49:d5:cc:a8:90:2b:7f:1b:58:7b:1f:26:92:18:4c:2d:37:65:5c
    9f:0f:6e:10:b5:34:6f:2d:b5:9c:27:3b:a6:b1:b5:a0:e2:a6:92:3d
    e4:68:fe:5d:71:06:6f:ce:e6:0f:0f:e3:94:2a:23:57:98:a0:6a:9c
    e0:52:a2:47:ff:ce:b0:47:bd:36:95:80:a7:af:d2:49:b1:5d:2a:3d
    28:e4:95:06:b8:b3:d9:07:11:3c:13:af:c6:e2:51:08:22:82:2d:ec
    4f:26:40:b0:b0:55:2d:6e:c0:c8:19:34:a7:99:5a:bc:58:98:69:ae
    04:d6:6d:ec:4a:c9:55:a5:ff:00:cb:3b:02:85:fa:02:a1:5c:c1:9d
    6d:44:b8:95:8f:77:c0:53:fc:7f:a4:09:a3:25:1c:4a:e2:9d:0c:81
    08:b4:c8:b8:0d:bc:94:75:54:75:57:4f:d3:a4:17:0d:5d:1a:f3:c1
    1d:5d:73:2f:fe:8b:cb:fc:1f:93:87:72:d6:be:df:86:d7:e6:d1:c7
    0d:00:1a:6e:58:db:6a:1c:2f:1d:17:46:9a:f2:69:b4:21:db:08:5d
    8d:ab:30:7d:7f:02:03:01:00:01
  Signature algorithm: sha256WithRSAEncryption
  Distribution CRL:
     http://10.102.40.55:8080/crl-as-der/currentcrl-11.crl?id=11
  Use for key: CRL signing, Certificate signing, Key encipherment, Digital signature
  Fingerprint:
    8b:84:60:2a:58:5b:80:f0:b9:ae:25:9f:67:3d:d6:81:ee:43:6c:d4 (sha1)
    ab:ec:4d:fe:d4:04:9c:c9:79:1d:9a:33:4e:6d:78:f6 (md5)

9d:f0:c0:a0:93:74:11:53:d3:4d:2d:75:d3:60:37:5f:fb:b7:a9:67:42:cd:7c:3c:0e:0f:9b:58:36:3c:14:f5
(sha256)
```

4

CHAPTER

IPsec VPN in Junos OS

Internet Key Exchange (IKE) for IPsec VPN

IN THIS SECTION

Internet Key Exchange version 2 (IKEv2) is an IPsec based tunneling protocol that provides a secure VPN communication channel between peer VPN devices and defines negotiation and authentication for IPsec security associations (SAs) in a protected manner.

IKE and IPsec Packet Processing

IN THIS SECTION

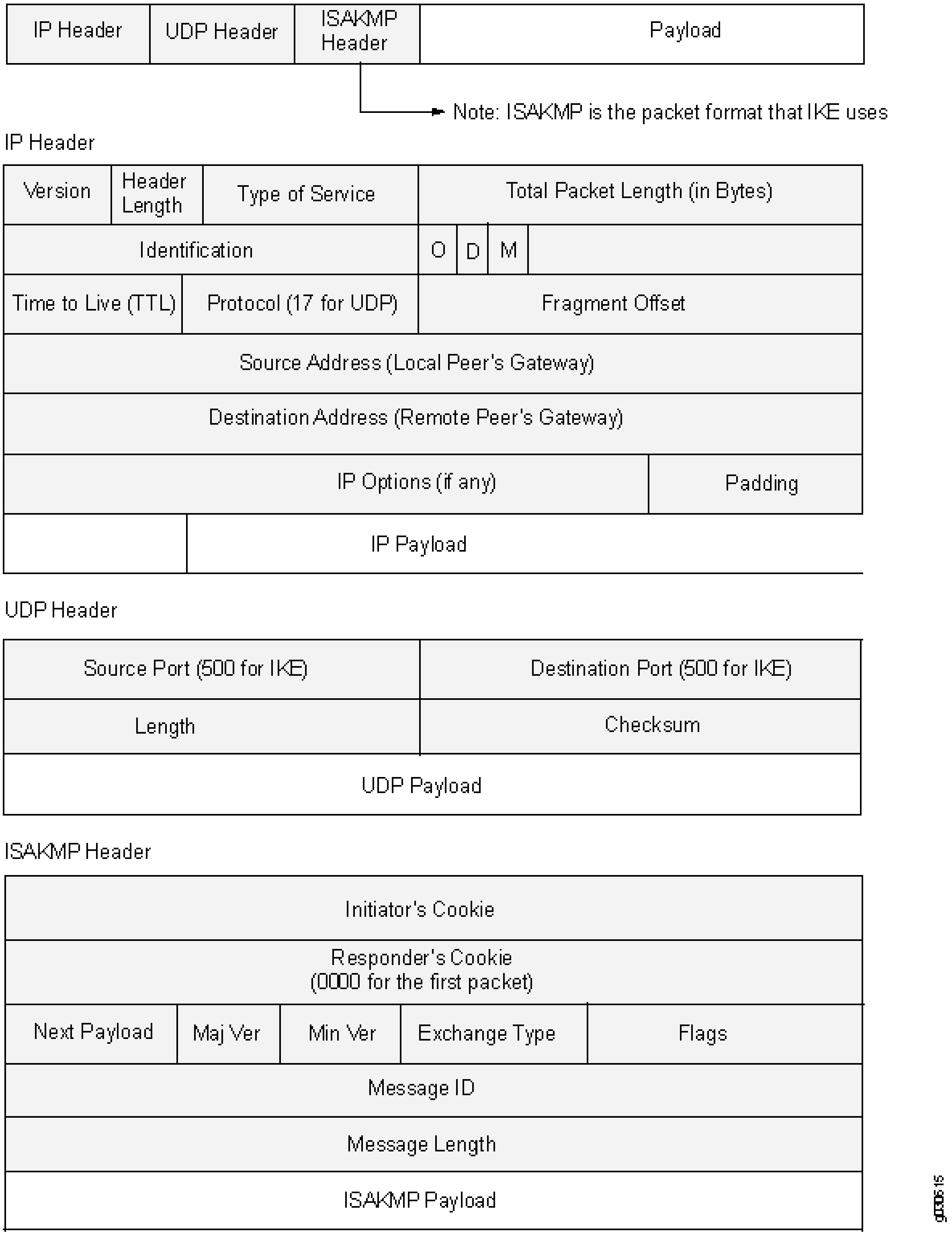
IKE provides tunnel management for IPsec and authenticates end entities. IKE performs a Diffie-Hellman (DH) key exchange to generate an IPsec tunnel between network devices. The IPsec tunnels generated by IKE are used to encrypt, decrypt, and authenticate user traffic between the network devices at the IP layer.

IKE Packet Processing

When a cleartext packet arrives on a Juniper Networks device that requires tunneling, and no active Phase 2 SA exists for that tunnel, Junos OS begins IKE negotiations and drops the packet. The source and destination addresses in the IP packet header are those of the local and remote IKE gateways, respectively. In the IP packet payload, there is a UDP segment encapsulating an ISAKMP (IKE) packet. The format for IKE packets is the same for Phase 1 and Phase 2. See Figure 12 on page 130.

Meanwhile, the source host has sent the dropped packet again. Typically, by the time the second packet arrives, IKE negotiations are complete, and Junos OS protects the packet and all subsequent packets in the session—with IPsec before forwarding it.

Figure 12: IKE Packet for Phases 1 and 2



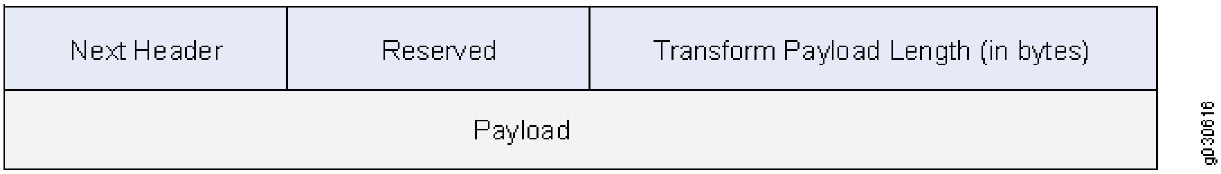
The Next Payload field contains a number indicating one of the following payload types:

- 0002—SA Negotiation Payload contains a definition for a Phase 1 or Phase 2 SA.
- 0004—Proposal Payload can be a Phase 1 or Phase 2 proposal.
- 0008—Transform Payload gets encapsulated in a proposal payload that gets encapsulated in an SA payload.
- 0010—Key Exchange (KE) Payload contains information necessary for performing a key exchange, such as a DH public value.
- 0020—Identification (IDx) Payload.
  - In Phase 1, IDii indicates the initiator ID, and IDir indicates the responder ID.
  - In Phase 2, IDui indicates the user initiator, and IDur indicates the user responder.

  The IDs are IKE ID types such as FQDN, U-FQDN, IP address, and ASN.1_DN.
- 0040—Certificate (CERT) Payload.
- 0080—Certificate Request (CERT_REQ) Payload.
- 0100—Hash (HASH) Payload contains the digest output of a particular hash function.
- 0200—Signature (SIG) Payload contains a digital signature.
- 0400—Nonce (Nx) Payload contains some pseudorandom information necessary for the exchange).
- 0800—Notify Payload.
- 1000—ISAKMP Delete Payload.
- 2000—Vendor ID (VID) Payload can be included anywhere in Phase 1 negotiations. Junos OS uses it to mark support for NAT-T.

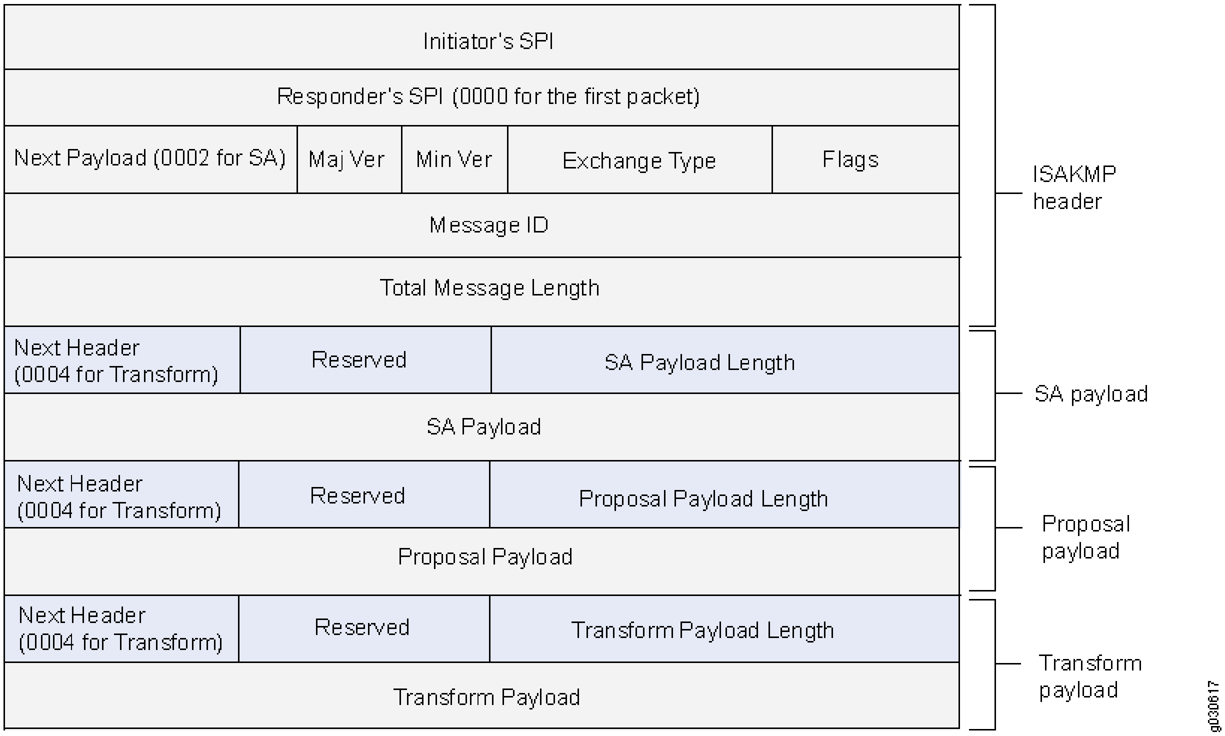
Each ISAKMP payload begins with the same generic header, as shown in Figure 13 on page 131.

Figure 13: Generic ISAKMP Payload Header



There can be multiple ISAKMP payloads chained together, with each subsequent payload type indicated by the value in the Next Header field. A value of **0000** indicates the last ISAKMP payload. See Figure 14 on page 132 for an example.

Figure 14: ISAKMP Header with Generic ISAKMP Payloads

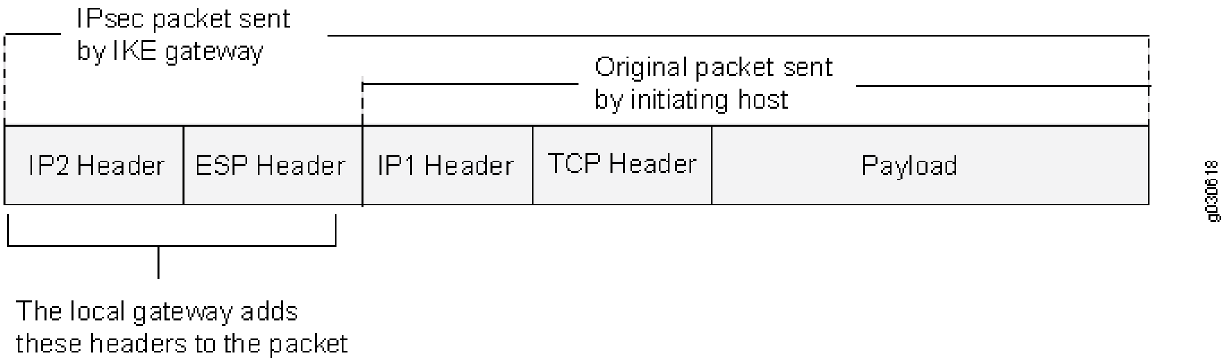


IPsec Packet Processing

After IKE negotiations complete and the two IKE gateways have established Phase 1 and Phase 2 security associations (SAs), all subsequent packets are forwarded using the tunnel. If the Phase 2 SA specifies the Encapsulating Security Protocol (ESP) in tunnel mode, the packet looks like the one shown in Figure 15 on page 133. The device adds two additional headers to the original packet that the initiating host sends.

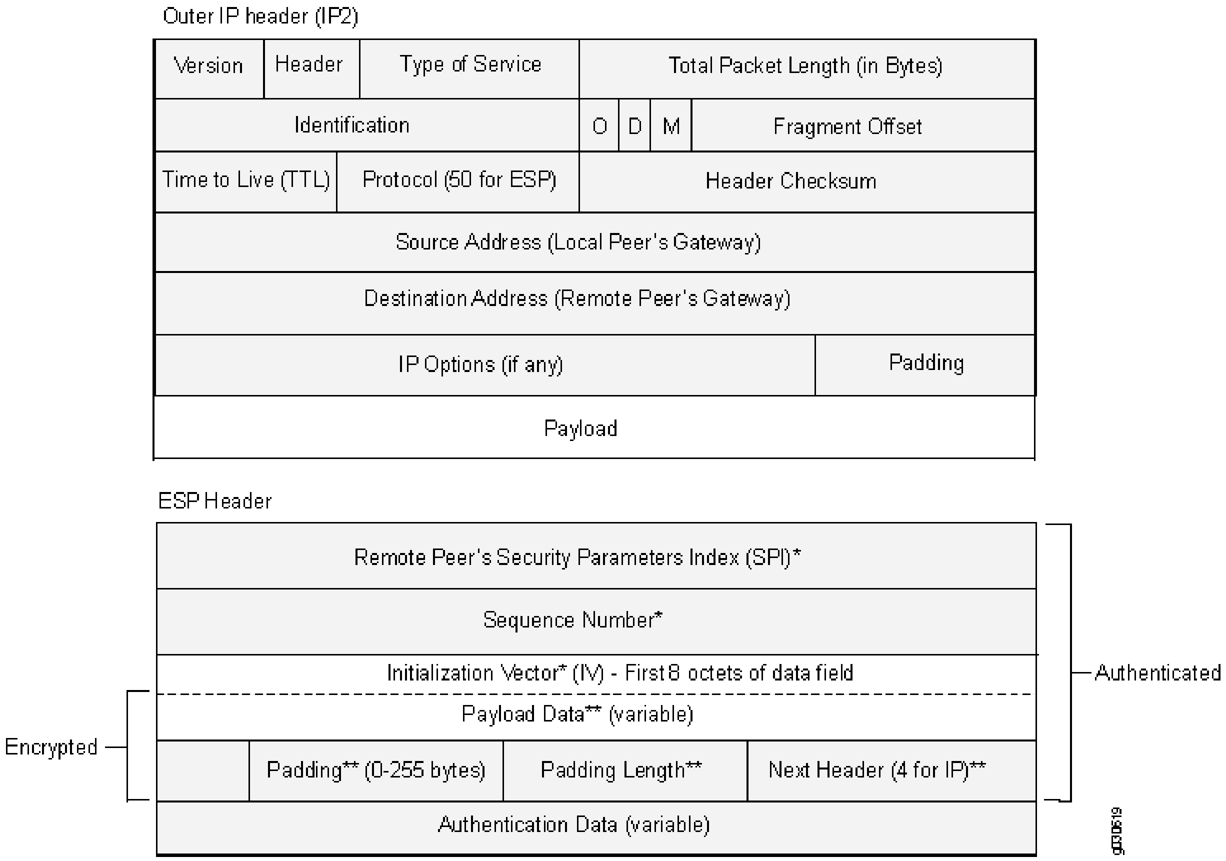
As shown in Figure 15 on page 133, the packet that the initiating host constructs includes the payload, the TCP header, and the inner IP header (IP1).

Figure 15: IPsec Packet—ESP in Tunnel Mode


IPsec packet sent by IKE gateway
Original packet sent by initiating host
IP2 Header
ESP Header
IP1 Header
TCP Header
Payload
g030618
The local gateway adds these headers to the packet

The router IP header (IP2), which Junos OS adds, contains the IP address of the remote gateway as the destination IP address and the IP address of the local router as the source IP address. Junos OS also adds an ESP header between the outer and inner IP headers. The ESP header contains information that allows the remote peer to properly process the packet when it receives it. This is shown in Figure 16 on page 134.

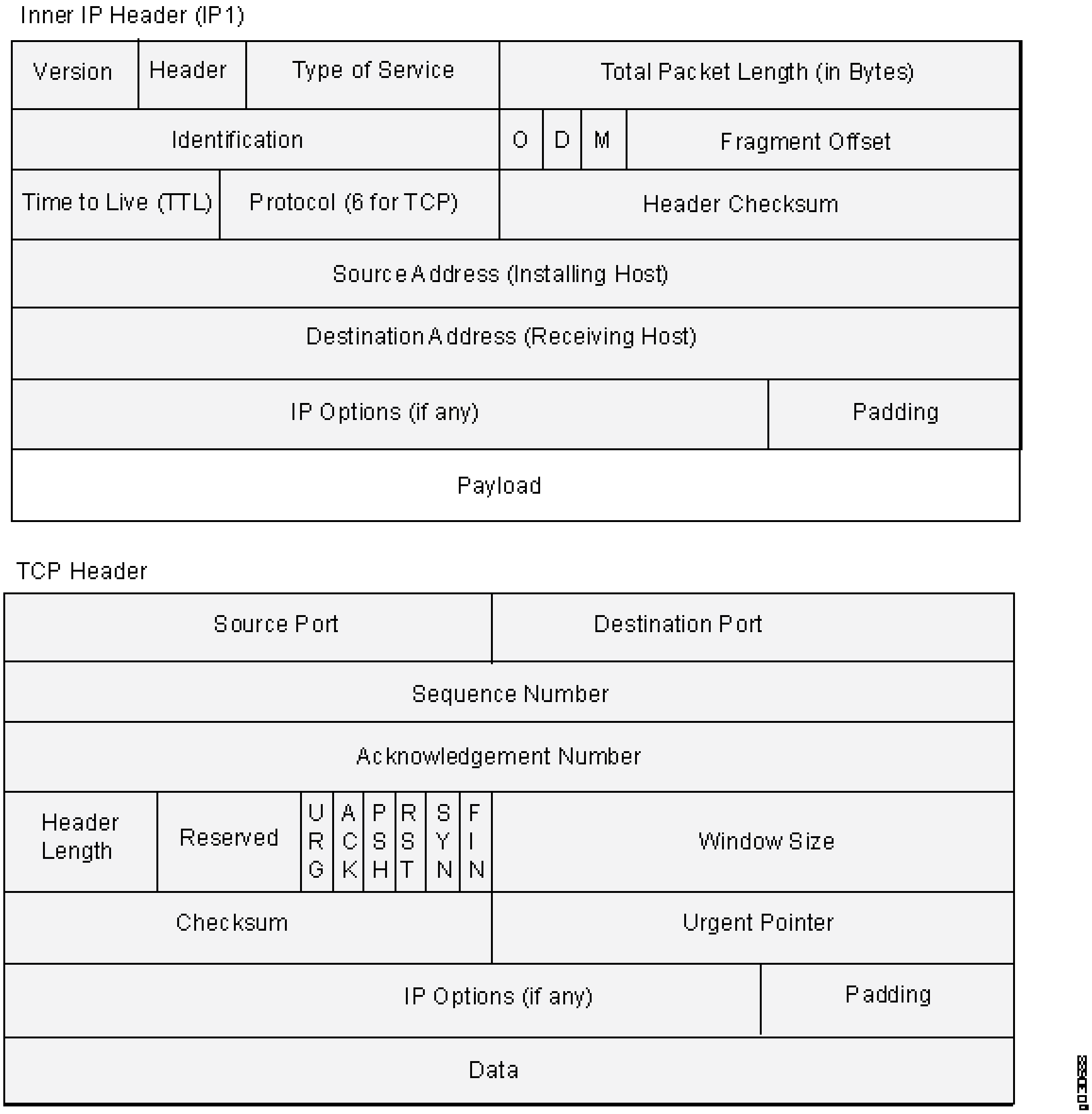
Figure 16: Outer IP Header (IP2) and ESP Header



* = Authenticated sections of the packet
** = Encrypted sections of the packet

The Next Header field indicates the type of data in the payload field. In tunnel mode, this value is 4, indicating an IP packet is contained within the payload. See Figure 17 on page 135.

Figure 17: Inner IP Header (IP1) and TCP Header



Introduction to IKE in Junos OS

IN THIS SECTION

IKE provides ways to exchange keys for encryption and authentication securely over an unsecured medium such as the Internet. IKE enables a pair of security gateways to: Dynamically establish a secure tunnel over which security gateways can exchange tunnel and key information. Set up user-level tunnels or SAs, including tunnel attribute negotiations and key management. These tunnels can also be refreshed and terminated on top of the same secure channel. IKE employs Diffie-Hellman methods and is optional in IPsec (the shared keys can be entered manually at the endpoints).

IKEv2 includes support for:

- Route-based VPNs.
- Site-to-site VPNs.
- Dead peer detection.
- *Chassis cluster*.
- Pre-shared key authentication.
- Certificate-based authentication.
- Child SAs. An IKEv2 child SA is known as a Phase 2 SA in IKEv1. In IKEv2, a child SA cannot exist without the underlying IKE SA.
- AutoVPN.
- Dynamic endpoint VPN.
- EAP is supported for Remote Access using IKEv2.
- Traffic selectors.

IKEv2 does not support the following features:

- Policy-based VPN.

- VPN monitoring.
- IP Payload Compression Protocol (IPComp).

Configuring IKEv2 in Junos OS

A VPN peer is configured as either IKEv1 or IKEv2. When a peer is configured as IKEv2, it cannot fall back to IKEv1 if its remote peer initiates IKEv1 negotiation. By default, Juniper Networks security devices are IKEv1 peers.

Use the `version v2-only` configuration statement at the `[edit security ike gateway` *gw-name*`]` hierarchy level to configure IKEv2.

The IKE version is displayed in the output of the `show security ike security-associations` and `show security ipsec security-associations` CLI operational commands.

Juniper Networks devices support up to four proposals for Phase 2 negotiations, allowing you to define how restrictive a range of tunnel parameters you will accept. Junos OS provides predefined standard, compatible, and basic Phase 2 proposal sets. You can also define custom Phase 2 proposals.

Understanding IKEv2 Configuration Payload

Configuration payload is an Internet Key Exchange version 2 (IKEv2) option offered to propagate provisioning information from a responder to an initiator. IKEv2 configuration payload is supported with route-based VPNs only.

RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*, defines 15 different configuration attributes that can be returned to the initiator by the responder. Table 5 on page 137 describes the IKEv2 configuration attributes supported on SRX Series Firewalls.

Table 5: IKEv2 Configuration Attributes

Attribute Type	Value	Description	Length
INTERNAL_IP4_ADDRESS	1	Specifies an address on the internal network. Multiple internal addresses can be requested. The responder can send up to the number of addresses requested.	0 or 4 octets
INTERNAL_IP4_NETMASK	2	Specifies the internal network's netmask value. Only one netmask value is allowed in the request and response messages (for example, 255.255.255.0), and it must be used only with an INTERNAL_IP4_ADDRESS attribute.	0 or 4 octets

Table 5: IKEv2 Configuration Attributes *(Continued)*

Attribute Type	Value	Description	Length
INTERNAL_IP4_DNS	3	Specifies an address of a DNS server within the network. Multiple DNS servers can be requested. The responder can respond with zero or more DNS server attributes.	0 or 4 octets
INTERNAL_IP4_NBNS	4	Specifies an address of a NetBIOS name server (NBNS), for example, a WINS server, within the network. Multiple NBNS servers can be requested. The responder can respond with zero or more NBNS server attributes.	0 or 4 octets
INTERNAL_IP6_ADDRESS	8	Specifies an address on the internal network. Multiple internal addresses can be requested. The responder can send up to the number of addresses requested.	0 or 17 octets
INTERNAL_IP6_DNS	10	Specifies an address of a DNS server within the network. Multiple DNS servers can be requested. The responder can respond with zero or more DNS server attributes.	0 or 16 octets

For the IKE responder to provide the initiator with provisioning information, it must acquire the information from a specified source such as a RADIUS server. Provisioning information can also be returned from a DHCP server through a RADIUS server. On the RADIUS server, the user information should not include an authentication password. The RADIUS server profile is bound to the IKE gateway using the `aaa` *`access-profile profile-name`* configuration at the [`edit security ike gateway` *`gateway-name`*] hierarchy level.

Starting in Junos OS Release 20.3R1, on SRX5000 line with SPC3 and vSRX Virtual Firewall running iked process, we've improved IKEv2 configuration payload to:

- Support for IPv4 and IPv6 local address pool. You can also assign a fixed IP address to a peer.

  During IKE establishment, the initiator requests for an IPv4 address, IPv6 address, DNS address, or WINS address from the responder. After the responder has authenticated the initiator successfully, it assigns an IP address either from a local address pool or through RADIUS server. Depending on the configuration, this IP address is either assigned dynamically each time when a peer connects or assigned as a fixed IP address. If the RADIUS server responds with a framed pool, Junos OS assigns an IP address or information based on configuration from it's corresponding local pool. If you configure both local address pool and RADIUS server, the IP address allocated from RADIUS server takes precedence over the local pool. If you configure local IP address pool and the RADIUS server did not return any IP address, then local pool assigns the IP address to the request.

- Additional option, none introduced for authentication-order. See *authentication-order (Access Profile)*.

- RADIUS accounting start and stop messages inform the state of the tunnel or peer to the RADIUS server. These messages can be used for tracking purposes or notifications to subsystems such as a DHCP server.

  Ensure that the RADIUS server support accounting start or stop messages. Also ensure that both the SRX Series Firewalls and the RADIUS server have appropriate settings to track these messages.

- Introduction of IPv6 support allows dual stack tunnels using configuration payload. During login process, IKE requests for both IPv4 and IPv6 addresses. AAA allow login only if all requested addresses have been allocated successfully. IKE terminates the negotiation if the requested IP is not allocated.

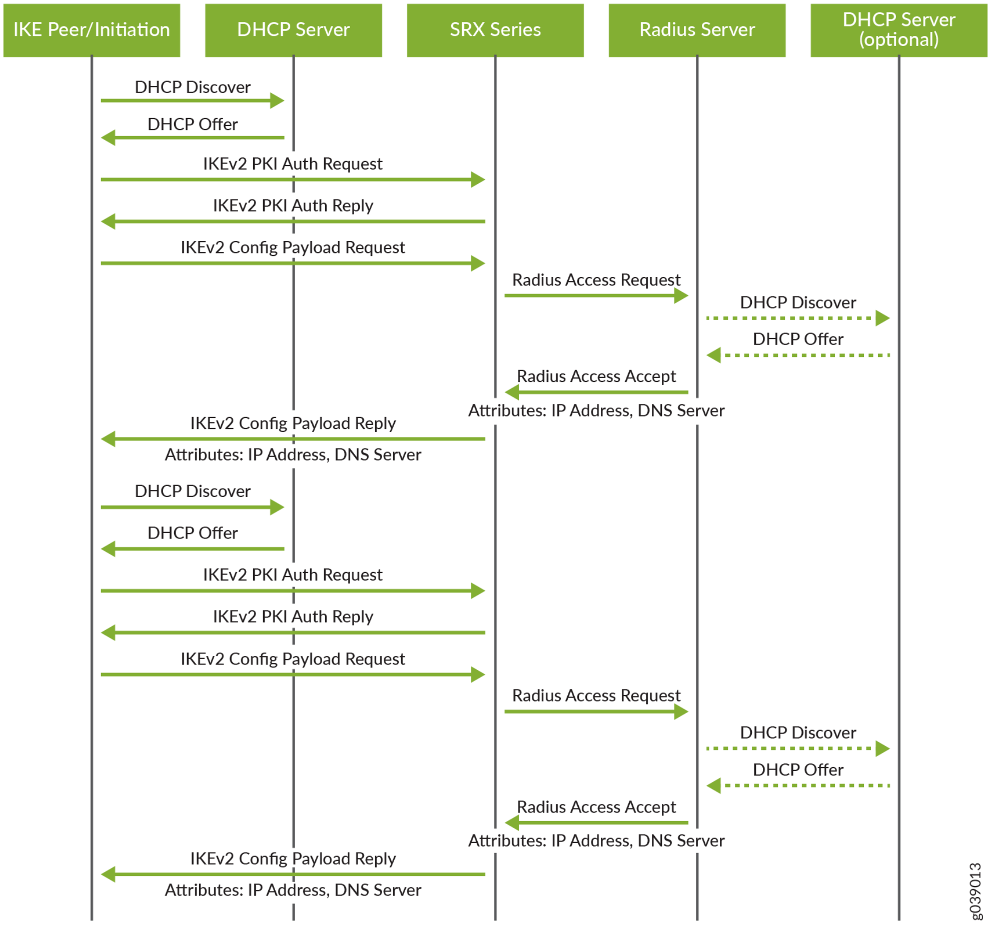
In a route-based VPN, secure tunnel (st0) interfaces operate in either point-to-multipoint or point-to-point mode. Address assignment through the IKEv2 configuration payload is now supported for point-to-multipoint or point-to-point mode. For point-to-multipoint interfaces, the interfaces must be numbered and the addresses in the configuration payload INTERNAL_IP4_ADDRESS attribute type must be within the subnetwork range of the associated point-to-multipoint interface.

Starting in Junos OS Release 20.1R1, you can configure a common password for IKEv2 configuration payload requests for an IKE gateway configuration. The common password in the range of 1 to 128 characters allows the administrator to define a common password. This password is used between the SRX Series Firewall and the RADIUS server when the SRX Series Firewall requesting an IP address on behalf of a remote IPsec peer using IKEv2 configuration payload. RADIUS server matches the credentials before it provides any IP information to the SRX Series Firewall for the configuration payload request. You can configure the common password using config-payload-password *configured-password* configuration statement at [edit security ike gateway gateway-name aaa access-profile access-profile-name] hierarchy level.

Both the SRX Series Firewall and the RADIUS server must have the same password configured and the radius server should be configured to use Password Authentication Protocol (PAP) as the authentication protocol. Without this, tunnel establishment will not be successful.

Figure 18 on page 140 shows a typical workflow for a IKEv2 Configuration Payload.

Figure 18: Typical IKEv2 Configuration Payload Workflow



The IKEv2 configuration payload feature is supported for both point-to-multipoint secure tunnel (st0) interfaces and point-to-point interfaces. Point-to-multipoint interfaces must be numbered, and the addresses provided in the configuration payload must be within the subnetwork range of the associated point-to-multipoint interface.

Starting in Junos OS Release 20.1R1, we support IKEv2 configuration payload feature with point-to-point interfaces on SRX5000 line and vSRX Virtual Firewall running iked.

Understanding Pico Cell Provisioning

IKEv2 configuration payload can be used to propagate provisioning information from an IKE responder, such as an SRX Series Firewall, to multiple initiators, such as LTE pico cell base stations in a cellular network. The pico cells ship from the factory with a standard configuration that allows them to connect to the SRX Series Firewall, but the pico cell provisioning information is stored on one or more provisioning servers within a protected network. The pico cells receive full provisioning information after establishing secure connections with the provisioning servers.

The workflow required to bootstrap and provision a pico cell and introduce it to service includes four distinct stages:

1. Initial addresses acquisition—The pico cell ships from the factory with the following information:

   - Configuration for the secure gateway tunnel to the SRX Series Firewall

   - Digital certificate issued by the manufacturer

   - Fully qualified domain name (FQDN) of the provisioning servers that lie within the protected network

   The pico cell boots up and acquires an address to be used for IKE negotiation from a DHCP server. A tunnel is then built to the secure gateway on the SRX Series Firewall using this address. An address for Operation, Administration, and Management (OAM) traffic is also assigned by the DHCP server for use on the protected network.

2. Pico cell provisioning—Using its assigned OAM traffic address, the pico cell requests its provisioning information—typically operator certificate, license, software, and configuration information—from servers within the protected network.

3. Reboot—The pico cell reboots and uses the acquired provisioning information to make it specific to the service provider’s network and operation model.

4. Service provision—When the pico cell enters service, it uses a single certificate that contains distinguished name (DN) and subject alternative name values with a FQDN to build two tunnels to the secure gateway on the SRX Series Firewall: one for OAM traffic and the other for Third-Generation Partnership Project (3GPP) data traffic.

SEE ALSO

Example: Configuring NAT-T with Dynamic Endpoint VPN | **678**

IKE Proposal

The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway. You can configure one or more IKE proposals. Each proposal is a list of IKE attributes to protect the IKE connection between the IKE host and its peer.

To configure an IKE proposal, include the `proposal` statement and specify a name at the [`edit security ike`] hierarchy level:

IKE Policy

An IKE policy defines a combination of security parameters (IKE proposals) to be used during IKE negotiation. It defines a peer address and the proposals needed for that connection. Depending on which authentication method is used, it defines the preshared key for the given peer or the local certificate. During the IKE negotiation, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match. A match is made when both policies from the two peers have a proposal that contains the same configured attributes. If the lifetimes are not identical, the shorter lifetime between the two policies (from the host and peer) is used. The configured preshared key must also match its peer.

First, you configure one or more IKE proposals; then you associate these proposals with an IKE policy.

To configure an IKE policy, include the `policy` statement and specify a policy name at the [`edit security ike`] hierarchy level:

Rekeying and Reauthentication

IN THIS SECTION

Overview

With IKEv2, rekeying and reauthentication are separate processes. Rekeying establishes new keys for the IKE security association (SA) and resets message ID counters, but it does not reauthenticate the peers. Reauthentication verifies that VPN peers retain their access to authentication credentials. Reauthentication establishes new keys for the IKE SA and child SAs; rekeys of any pending IKE SA or child SA are no longer needed. After the new IKE and child SAs are created, the old IKE and child SAs are deleted.

IKEv2 reauthentication is disabled by default. You enable reauthentication by configuring a reauthentication frequency value between 1 and 100. The reauthentication frequency is the number of IKE rekeys that occurs before reauthentication occurs. For example, if the configured reauthentication frequency is 1, reauthentication occurs every time there is an IKE rekey. If the configured reauthentication frequency is 2, reauthentication occurs at every other IKE rekey. If the configured reauthentication frequency is 3, reauthentication occurs at every third IKE rekey, and so on.

You configure the reauthentication frequency with the `reauth-frequency` statement at the [`edit security ike policy` *`policy-name`*] hierarchy level. Reauthentication is disabled by setting the reauthentication frequency to 0 (the default). Reauthentication frequency is not negotiated by peers, and each peer can have its own reauthentication frequency value.

Supported Features

IKEv2 reauthentication is supported with the following features:

- IKEv2 initiators or responders
- Dead peer detection (DPD)
- Virtual routers and secure tunnel (st0) interfaces in virtual routers
- Network Address Translation traversal (NAT-T)
- Chassis clusters in active-active and active-passive mode for SRX5400, SRX5600, and SRX5800 devices
- In-service software upgrade (ISSU) on SRX5400, SRX5600, and SRX5800 devices
- Upgrade or insertion of a new Services Processing Unit (SPU) using the in-service hardware upgrade (ISHU) procedure

Limitations

Note the following caveats when using IKEv2 reauthentication:

- With NAT-T, a new IKE SA can be created with different ports from the previous IKE SA. In this scenario, the old IKE SA might not be deleted.
- In a NAT-T scenario, the initiator behind the NAT device can become the responder after reauthentication. If the NAT session expires, the NAT device might discard new IKE packets that might arrive on a different port. NAT-T keepalive or DPD must be enabled to keep the NAT session alive. For AutoVPN, we recommend that the reauthentication frequency configured on the spokes be smaller than the reauthentication frequency configured on the hub.

- Based on the reauthentication frequency, a new IKE SA can be initiated by either the initiator or the responder of the original IKE SA. Because Extensible Authentication Protocol (EAP) authentication and configuration payload require the IKE SA to be initiated by the same party as the original IKE SA, reauthentication is not supported with EAP authentication or configuration payload.

IKE Authentication (Certificate-Based Authentication)

IN THIS SECTION

Multilevel Hierarchy for Certificate Authentication

Certificate-based authentication is an authentication method supported on SRX Series Firewalls during IKE negotiation. In large networks, multiple certificate authorities (CAs) can issue end entity (EE) certificates to their respective end devices. It is common to have separate CAs for individual locations, departments, or organizations.

When a single-level hierarchy for certificate-based authentication is employed, all EE certificates in the network must be signed by the same CA. All firewall devices must have the same CA certificate enrolled for peer certificate validation. The certificate payload sent during IKE negotiation only contains EE certificates.

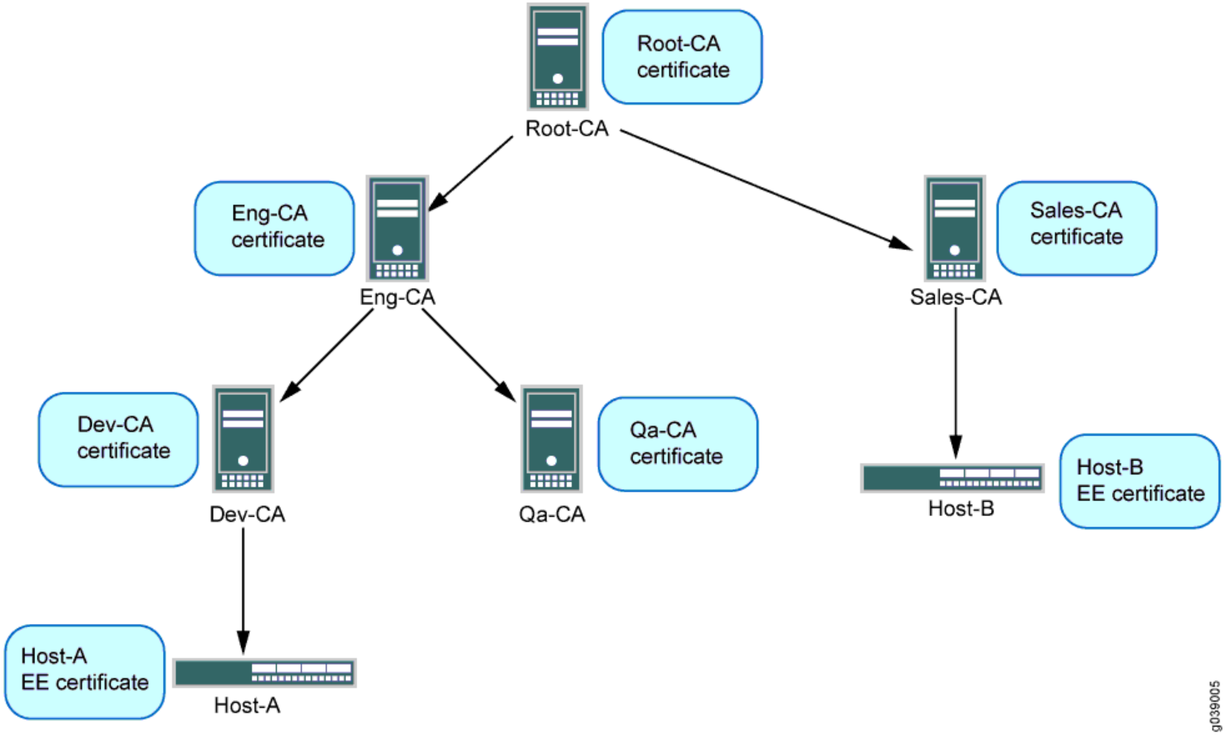
Alternatively, the certificate payload sent during IKE negotiation can contain a chain of EE and CA certificates. A *certificate chain* is the list of certificates required to validate a peer's EE certificate. The certificate chain includes the EE certificate and any CA certificates that are not present in the local peer.

The network administrator needs to ensure that all peers participating in an IKE negotiation have at least one common trusted CA in their respective certificate chains. The common trusted CA does not have to be the root CA. The number of certificates in the chain, including certificates for EEs and the topmost CA in the chain, cannot exceed 10.

Starting with Junos OS Release 18.1R1, validation of a configured IKE peer can be done with a specified CA server or group of CA servers. With certificate chains, the root CA must match the trusted CA group or CA server configured in the IKE policy

In the example CA hierarchy shown in Figure 19 on page 145, Root-CA is the common trusted CA for all devices in the network. Root-CA issues CA certificates to the engineering and sales CAs, which are identified as Eng-CA and Sales-CA, respectively. Eng-CA issues CA certificates to the development and quality assurance CAs, which are identified as Dev-CA and Qa-CA, respectively. Host-A receives its EE certificate from Dev-CA while Host-B receives its EE certificate from Sales-CA.

Figure 19: Multilevel Hierarchy for Certificate-Based Authentication



Each end device needs to be loaded with the CA certificates in its hierarchy. Host-A must have Root-CA, Eng-CA, and Dev-CA certificates; Sales-CA and Qa-CA certificates are not necessary. Host-B must have Root-CA and Sales-CA certificates. Certificates can be loaded manually in a device or enrolled using the Simple Certificate Enrollment Process (SCEP).

Each end device must be configured with a CA profile for each CA in the certificate chain. The following output shows the CA profiles configured on Host-A:

```
admin@host-A# show security
pki {
    ca-profile Root-CA {
        ca-identity Root-CA;
        enrollment {
            url “www.example.net/scep/Root/”;
        }
    }
    ca-profile Eng-CA {
        ca-identity Eng-CA;
        enrollment {
            url “www.example.net/scep/Eng/”;
        }
    }
    ca-profile Dev-CA {
        ca-identity Dev-CA;
        enrollment {
            url “www.example.net/scep/Dev/”;
        }
    }
}
```

The following output shows the CA profiles configured on Host-B:

```
admin@host-B# show security
pki {
    ca-profile Root-CA {
        ca-identity Root-CA;
        enrollment {
            url “www.example.net/scep/Root/”;
        }
    }
    ca-profile Sales-CA {
        ca-identity Sales-CA;
        enrollment {
            url “www.example.net/scep/Sales/”;
        }
    }
}
```

SEE ALSO

- Basic Elements of PKI in Junos OS
- Understanding Certificate Authority Profiles

Example: Configuring a Device for Peer Certificate Chain Validation

IN THIS SECTION

This example shows how to configure a device for certificate chains used to validate peer devices during IKE negotiation.

Requirements

Before you begin, obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

Overview

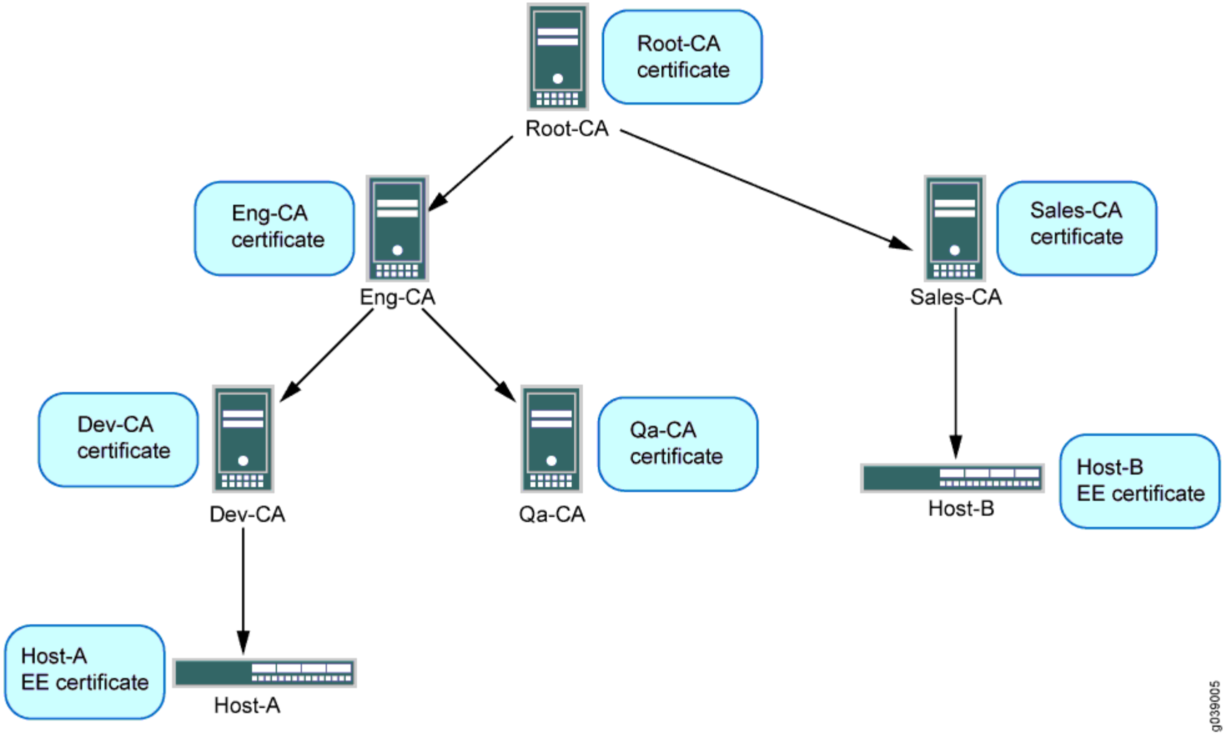
IN THIS SECTION

This example shows how to configure a local device for certificate chains, enroll CA and local certificates, check the validity of enrolled certificates, and check the revocation status of the peer device.

Topology

This example shows the configuration and operational commands on Host-A, as shown in Figure 20 on page 148. A dynamic CA profile is automatically created on Host-A to allow Host-A to download the CRL from Sales-CA and check the revocation status of Host-B’s certificate.

Figure 20: Certificate Chain Example



The IPsec VPN configuration for Phase 1 and Phase 2 negotiation is shown for Host-A in this example. The peer device (Host-B) must be properly configured so that Phase 1 and Phase 2 options are successfully negotiated and security associations (SAs) are established. See "Configuring Remote IKE IDs for Site-to-Site VPNs" on page 199 for examples of configuring peer devices for VPNs.

Configuration

IN THIS SECTION

To configure a device for certificate chains:

Configure CA Profiles

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile Root-CA ca-identity CA-Root
set security pki ca-profile Root-CA enrollment url http://198.51.100.230:8080/scep/Root/
set security pki ca-profile Root-CA revocation-check crl
set security pki ca-profile Eng-CA ca-identity Eng-CA
set security pki ca-profile Eng-CA enrollment url http://198.51.100.230:8080/scep/Eng/
set security pki ca-profile Eng-CA revocation-check crl
set security pki ca-profile Dev-CA ca-identity Dev-CA
set security pki ca-profile Dev-CA enrollment url http://198.51.100.230:8080/scep/Dev/
set security pki ca-profile Dev-CA revocation-check crl
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure CA profiles:

1. Create the CA profile for Root-CA.

   ```
   [edit security pki]
   user@host# set ca-profile Root-CA ca-identity CA-Root
   user@host# set ca-profile Root-CA enrollment url http://198.51.100.230:8080/scep/Root/
   user@host# set ca-profile Root-CA revocation-check crl
   ```

2. Create the CA profile for Eng-CA.

   ```
   [edit security pki]
   user@host# set ca-profile Eng-CA ca-identity Eng-CA
   user@host# set ca-profile Eng-CA enrollment url http://198.51.100.230:8080/scep/Eng/
   user@host# set ca-profile Eng-CA revocation-check crl
   ```

3. Create the CA profile for Dev-CA.

```
[edit security pki]
user@host# set ca-profile Dev-CA ca-identity Dev-CA
user@host# set ca-profile Dev-CA enrollment url http://198.51.100.230:8080/scep/Dev/
user@host# set ca-profile Dev-CA revocation-check crl
```

Results

From configuration mode, confirm your configuration by entering the `show security pki` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security pki
ca-profile Root-CA {
    ca-identity Root-CA;
    enrollment {
        url "http:/;/198.51.100.230:8080/scep/Root/";
    }
    revocation-check {
        crl ;
    }
}
ca-profile Eng-CA {
    ca-identity Eng-CA;
    enrollment {
        url "http:/;/198.51.100.230:8080/scep/Eng/";
    }
    revocation-check {
        crl ;
    }
}
ca-profile Dev-CA {
    ca-identity Dev-CA;
    enrollment {
        url "http:/;/198.51.100.230:8080/scep/Dev/";
    }
    revocation-check {
        crl ;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Enroll Certificates

Step-by-Step Procedure

To enroll certificates:

1. Enroll the CA certificates.

   ```
   user@host> request security pki ca-certificate enroll ca-profile Root-CA
   ```

   ```
   user@host> request security pki ca-certificate enroll ca-profile Eng-CA
   ```

   ```
   user@host> request security pki ca-certificate enroll ca-profile Dev-CA
   ```

   Type **yes** at the prompts to load the CA certificate.

2. Verify that the CA certificates are enrolled in the device.

   ```
   user@host> show security pki ca-certificate ca-profile Root-CA
   Certificate identifier: Root-CA
           Issued to: Root-CA, Issued by: C = us, O = example, CN = Root-CA
           Validity:
             Not before: 08-14-2012 22:19
             Not after: 08-13-2017 22:19
           Public key algorithm: rsaEncryption(2048 bits)
   ```

   ```
   user@host> show security pki ca-certificate ca-profile Eng-CA
   Certificate identifier: Eng-CA
           Issued to: Eng-CA, Issued by: C = us, O = example, CN = Root-CA
           Validity:
             Not before: 08-15-2012 01:02
   ```

```
          Not after: 08-13-2017 22:19
        Public key algorithm: rsaEncryption(2048 bits)
```

```
user@host> show security pki ca-certificate ca-profile Dev-CA
        Certificate identifier: Dev-CA
        Issued to: Dev-CA, Issued by: C = us, O = example, CN = Eng-CA
        Validity:
          Not before: 08-15-2012 17:41
          Not after: 08-13-2017 22:19
        Public key algorithm: rsaEncryption(2048 bits)
```

3. Verify the validity of the enrolled CA certificates.

```
user@host> request security pki ca-certificate verify ca-profile Root-CA
CA certificate Root-CA verified successfully
```

```
user@host> request security pki ca-certificate verify ca-profile Eng-CA
CA certificate Eng-CA verified successfully
```

```
user@host> request security pki ca-certificate verify ca-profile Dev-CA
CA certificate Dev-CA verified successfully
```

4. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Host-A type rsa size 1024
```

5. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll certificate-id        Host-A ca-
profile Dev-CA challenge-password example domain-name host-a.example.net email host-
a@example.net subject DC=example,CN=Host-A,       OU=DEV,O=PKI,L=Sunnyvale,ST=CA,C=US
```

6. Verify that the local certificate is enrolled in the device.

```
user@host> show security pki local-certificate
Issued to: Host-A, Issued by: C = us, O = example, CN = Dev-CA
        Validity:
          Not before: 09-17-2012 22:22
          Not after: 08-13-2017 22:19
        Public key algorithm: rsaEncryption(1024 bits)
```

7. Verify the validity of the enrolled local certificate.

```
user@host> request security pki local-certificate verify certificate-id Host-A
Local certificate Host-A verification success
```

8. Check the CRL download for configured CA profiles.

```
user@host> show security pki crl
     CA profile: Root-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Root-CA
        Effective date: 09- 9-2012 13:08
        Next update: 09-21-2012 02:55

      CA profile: Eng-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Eng-CA
        Effective date: 08-22-2012 17:46
        Next update: 10-24-2015 03:33

      CA profile: Dev-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Dev-CA
        Effective date: 09-14-2012 21:15
        Next update: 09-26-2012 11:02
```

Configure IPsec VPN Options

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike_cert_prop_01 authentication-method rsa-signatures
set security ike proposal ike_cert_prop_01 dh-group group5
set security ike proposal ike_cert_prop_01 authentication-algorithm sha1
set security ike proposal ike_cert_prop_01 encryption-algorithm aes-256-cbc
set security ike policy ike_cert_pol_01 mode main
set security ike policy ike_cert_pol_01 proposals ike_cert_prop_01
set security ike policy ike_cert_pol_01 certificate local-certificate Host-A
set security ike gateway ike_cert_gw_01 ike-policy ike_cert_pol_01
set security ike gateway ike_cert_gw_01 address 192.0.2.51
set security ike gateway ike_cert_gw_01 external-interface ge-0/0/1.0
set security ike gateway ike_cert_gw_01 local-identity 192.0.2.31
set security ipsec proposal ipsec_prop_01 protocol esp
set security ipsec proposal ipsec_prop_01 authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_prop_01 encryption-algorithm 3des-cbc
set security ipsec proposal ipsec_prop_01 lifetime-seconds 300
set security ipsec policy ipsec_pol_01 proposals ipsec_prop_01
set security ipsec vpn ipsec_cert_vpn_01 bind-interface st0.1
set security ipsec vpn ipsec_cert_vpn_01 ike gateway ike_cert_gw_01
set security ipsec vpn ipsec_cert_vpn_01 ike ipsec-policy ipsec_pol_01
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec VPN options:

1. Configure Phase 1 options.

   ```
   [edit security ike proposal ike_cert_prop_01]
   user@host# set authentication-method rsa-signatures
   user@host# set dh-group group5
   user@host# set authentication-algorithm sha1
   ```

```
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy ike_cert_pol_01]
user@host# set mode main
user@host# set proposals ike_cert_prop_01
user@host# set certificate local-certificate Host-A
[edit security ike gateway ike_cert_gw_01]
user@host# set ike-policy ike_cert_pol_01
user@host# set address 192.0.2.51
user@host# set external-interface ge-0/0/1.0
user@host# set local-identity 192.0.2.31
```

2. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec_prop_01]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm 3des-cbc
user@host# set lifetime-seconds 300
[edit security ipsec policy ipsec_pol_01]
user@host# set proposals ipsec_prop_01
[edit security ipsec vpn ipsec_cert_vpn_01]
user@host# set bind-interface st0.1
user@host# set ike gateway ike_cert_gw_01
user@host# set ike ipsec-policy ipsec_pol_01
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` and `show security ipsec` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
proposal ike_cert_prop_01 {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
    policy ike_cert_pol_01 {
        mode main;
        proposals ike_cert_prop_01;
        certificate {
            local-certificate Host-A;
        }
    }
    gateway ike_cert_gw_01 {
        ike-policy ike_cert_pol_01;
        address 192.0.2.51;
        external-interface ge-0/0/1.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec_prop_01 {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 300;
}
    policy ipsec_pol_01 {
        proposals ipsec_prop_01;
    }
    vpn ipsec_cert_vpn_01 {
        bind-interface st0.1;
        ike {
            gateway ike_cert_gw_01;
            ipsec-policy ipsec_pol_01;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

If certificate validation is successful during IKE negotiation between peer devices, both IKE and IPsec security associations (SAs) are established.

The IKE SA is UP if the certificate is valid. The IKE SA is DOWN and IPSEC SA is formed if the certificate is revoked, only if revocation check is configured on the peer device

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

Enter the **show security ike security-associations** command from operational mode.

```
user@host> show security ike security-associations
    Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
    2090205 DOWN     285feacb50824495  59fca3f72b64da10  Main           192.0.2.51
```

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

Enter the **show security ipsec security-associations** command from operational mode.

```
user@host> show security ipsec security-associations
      Total active tunnels: 1
      ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
      <131073 ESP:3des/sha1 a4756de9 207/  unlim   -   root 500   192.0.2.51
      >131073 ESP:3des/sha1 353bacd3 207/  unlim   -   root 500   192.0.2.51
```

IKE and IPsec SA Failure for a Revoked Certificate

IN THIS SECTION

Checking for Revoked Certificates

Problem

If certificate validation fails during IKE negotiation between peer devices, check to make sure that the peer's certificate has not been revoked. A dynamic CA profile allows the local device to download the CRL from the peer's CA and check the revocation status of the peer's certificate. To enable dynamic CA profiles, the `revocation-check crl` option must be configured on a parent CA profile.

Solution

To check the revocation status of a peer's certificate:

1. Identify the dynamic CA profile that will show the CRL for the peer device by entering the **show security pki crl** command from operational mode.

```
user@host> show security pki crl
     CA profile: Root-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Root-CA
        Effective date: 09- 9-2012 13:08
        Next update: 09-21-2012 02:55

      CA profile: Eng-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Eng-CA
        Effective date: 08-22-2012 17:46
        Next update: 10-24-2015 03:33

      CA profile: Dev-CA
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Dev-CA
        Effective date: 09-14-2012 21:15
        Next update: 09-26-2012 11:02

      CA profile: dynamic-001
        CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Sales-CA
        Effective date: 09-14-2012 21:15
        Next update: 09-26-2012 11:02
```

The CA profile `dynamic-001` is automatically created on Host-A so that Host-A can download the CRL from Host-B’s CA (Sales-CA) and check the revocation status of the peer’s certificate.

2. Display CRL information for the dynamic CA profile by entering the **show security pki crl ca-profile dynamic-001 detail** command from operational mode.

   Enter

```
user@host> show security pki crl ca-profile dynamic-001 detail
    CA profile: dynamic-001
      CRL version: V00000001
        CRL issuer: C = us, O = example, CN = Sub11
        Effective date: 09-19-2012 17:29
        Next update: 09-20-2012 01:49
        Revocation List:
          Serial number              Revocation date
          10647C84                   09-19-2012 17:29 UTC
```

   Host-B’s certificate (serial number 10647084) has been revoked.

SEE ALSO

PKI Components In Junos OS | 33

Understanding Certificate Authority Profiles | 46

IKEv2 Fragmentation

IN THIS SECTION

Message Fragmentation

IKEv2 message fragmentation, as described in RFC 7383, *Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation*, allows IKEv2 to operate in environments where IP fragments might be blocked and peers would not be able to establish an IPsec security association (SA). IKEv2 fragmentation splits a large IKEv2 message into a set of smaller ones so that there is no fragmentation at the IP level. Fragmentation takes place before the original message is encrypted and authenticated, so that each fragment is separately encrypted and authenticated. On the receiver, the fragments are collected, verified, decrypted, and merged into the original message.

For IKEv2 fragmentation to occur, both VPN peers *must* indicate fragmentation support by including the IKEV2_FRAGMENTATION_SUPPORTED notification payload in the IKE_SA_INIT exchange. If both peers indicate fragmentation support, it is up to the initiator of the message exchange to determine whether or not IKEv2 fragmentation is used.

On SRX Series Firewalls, a maximum of 32 fragments are allowed per IKEv2 message. If the number of IKEv2 message fragments to be sent or received exceeds 32, the fragments are dropped and the tunnel is not established. Retransmission of individual message fragments is not supported

Configuration

On SRX Series Firewalls, IKEv2 fragmentation is enabled by default for IPv4 and IPv6 messages. To disable IKEv2 fragmentation, use the `disable` statement at the [`edit security ike gateway gateway-name fragmentation`] hierarchy level. You can also use the `size` statement to configure the size of the packet at which messages are fragmented; the packet size ranges from 500 to 1300 bytes. If `size` is not configured, the default packet size is 576 bytes for IPv4 traffic and 1280 bytes for IPv6 traffic. An IKEv2 packet that is larger than the configured packet size is fragmented.

After IKEv2 fragmentation is disabled or enabled or the packet fragment size is changed, the VPN tunnels that are hosted on the IKE gateway are brought down and IKE and IPsec SAs are renegotiated.

Caveats

The following features are not supported with IKEv2 fragmentation:

- Path MTU Discovery.
- SNMP.

SEE ALSO

Understanding Certificate Authority Profiles | 46

IKE Policy with a Trusted CA

This example shows how to bind a trusted CA server to an IKE policy of the peer.

Before you begin, you must have a list of all the trusted CAs you want to associate with the IKE policy of the peer.

You can associate an IKE policy to a single trusted CA profile or a trusted CA group. For establishing a secure connection, the IKE gateway uses the IKE policy to limit itself to the configured group of CAs (ca-profiles) while validating the certificate. A certificate issued by any source other than the trusted CA or trusted CA group is not validated. If there is a certificate validation request coming from an IKE policy then the associated CA profile of the IKE policy will validate the certificate. If an IKE policy is not associated with any CA then by default the certificate is validated by any one of the configured CA profiles.

In this example, a CA profile named `root-ca` is created and a `root-ca-identity` is associated to the profile.

You can configure a maximum of 20 CA profiles that you want to add to a trusted CA group. You cannot commit your configuration if you configure more than 20 CA profiles in a trusted CA group.

1. Create a CA profile and associate a CA identifier to the profile.

```
[edit]
user@host# set security pki ca-profile root-ca ca-identity root-ca
```

2. Define an IKE proposal and the IKE proposal authentication method.

```
[edit]
user@host# set security ike proposal ike_prop authentication-method rsa-signatures
```

3. Define the Diffie-Hellman group, authentication algorithm, an encryption algorithm for the IKE proposal.

```
[edit]
user@host# set security ike proposal ike_prop dh-group group2
user@host# set security ike proposal ike_prop authentication-algorithm sha-256
user@host# set security ike proposal ike_prop encryption-algorithm aes-256-cbc
```

4. Configure an IKE policy and associate the policy with the IKE proposal.

```
[edit]
user@host# set security ike policy ike_policy proposals ike_prop
```

5. Configure a local certificate identifier for the IKE policy.

```
[edit]
user@host# set security ike policy ike_policy certificate local-certificate SPOKE
```

6. Define the CA to be used for the IKE policy.

```
[edit]
user@host# set security ike policy ike_policy certificate trusted-ca ca-profile root-ca
```

To view the CA profiles and the trusted CA groups configured on your device, run `show security pki` command.

```
user@host# show security ike
    proposal ike_prop {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
}
policy ike_policy {
    proposals ike_prop;
    certificate {
        local-certificate SPOKE;
        trusted-ca ca-profile root-ca;
    }
}
```

The `show security ike` command displays the CA profile group under the IKE policy named `ike_policy` and the certificate associated with the IKE policy.

SEE ALSO

Understanding Certificate Authority Profiles | 46

Configuring Establish-Tunnel Responder-only in IKE

This topic shows how to configure establish-tunnels responder-only in Internet Key Exchange (IKE). Initiate the tunnels from the remote peer and send the traffic through all the tunnels. Specifies when IKE is activated.

Starting in Junos OS Release 19.1R1, on SRX5000 line, the establish tunnels option supports the `responder-only` and `responder-only-no-rekey` values under the `[edit security ipsec vpn` *`vpn-name`*`]` hierarchy-level.

The `responder-only` and `responder-only-no-rekey` options are supported on the SRX5000 line with an SPC3 card only if the `junos-ike-package` is installed. These options are supported only on a site-to-site VPN. These option are not supported on Auto VPN.

The `responder-only` and `responder-only-no-rekey` options does not establish any VPN tunnel from the device, so the VPN tunnel is initiated from the remote peer. When you configure `responder-only`, an established tunnel rekeys both IKE and IPsec based on the configured IKE and IPsec lifetime values. When you configure `responder-only-no-rekey`, an established tunnel does not rekey from the device and relies on the remote peer to initiate rekey. If the remote peer does not initiate rekey, then the tunnel teardown occurs after hard-lifetime expires.

Before you begin:

- Understand how to establish an AutoKey IKE IPsec tunnel. Read "IPsec Overview" on page 20.

To configure establish-tunnel responder-only in IKE:

1. Configure establish-tunnel responder-only

   ```
   user@host# set security ipsec vpn S2S_VPN establish-tunnel responder-only
   ```

2. Confirm your configuration by entering the `show security ipsec vpn IPSEC_VPN` command.

   ```
   user@host# show security ipsec vpn IPSEC_VPN
   bind-interface st0.1;
   ike {
           gateway IKE_GW;
           ipsec-policy IPSEC_POL;
       }
   establish-tunnels responder-only;
   ```

3. Configure establish-tunnel responder-only-no-rekey

   ```
   user@host# set security ipsec vpn S2S_VPN establish-tunnel responder-only-no-rekey
   ```

4. Confirm your configuration by entering the `show security ipsec vpn IPSEC_VPN` command.

   ```
   user@host# show security ipsec vpn IPSEC_VPN
   bind-interface st0.1;
   ike {
           gateway IKE_GW;
           ipsec-policy IPSEC_POL;
       }
   establish-tunnels responder-only-no-rekey;
   ```

   In case of multiple VPN objects, the Responder-only mode will take precedence. If any of the VPN in a gateway is configured with responder-only mode, all VPN's in the gateway must be configured with the responder-only mode.

Release History Table

Release	Description
18.1R1	Starting with Junos OS Release 18.1R1, validation of a configured IKE peer can be done with a specified CA server or group of CA servers.

RELATED DOCUMENTATION

Certificate Authority | 42

IPsec VPN Overview

IN THIS SECTION

A VPN is a private network that uses a public network to connect two or more remote sites. Instead of using dedicated connections between networks, VPNs use virtual connections routed (tunneled) through public networks. IPsec VPN is a protocol, consists of set of standards used to establish a VPN connection.

A VPN provides a means by which remote computers communicate securely across a public WAN such as the Internet.

A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. To secure VPN communication while passing through the WAN, the two participants create an IP Security (IPsec) tunnel.

The term *tunnel* does not denote tunnel mode (see "Packet Processing in Tunnel Mode" on page 170). Instead, it refers to the IPsec connection.

IPsec VPN Topologies on SRX Series Firewalls

The following are some of the IPsec VPN topologies that Junos operating system (OS) supports:

- Site-to-site VPNs—Connects two sites in an organization together and allows secure communications between the sites.
- Hub-and-spoke VPNs—Connects branch offices to the corporate office in an enterprise network. You can also use this topology to connect spokes together by sending traffic through the hub.
- Remote access VPNs—Allows users working at home or traveling to connect to the corporate office and its resources. This topology is sometimes referred to as an *end-to-site tunnel*.

SEE ALSO

Example: Configuring a Hub-and-Spoke VPN | **209**

Comparing Policy-Based and Route-Based VPNs

It is important to understand the differences between policy-based and route-based VPNs and why one might be preferable to the other.

Table 6 on page 166 lists the differences between route-based VPNs and policy-based VPNs.

Table 6: Differences Between Route-Based VPNs and Policy-Based VPNs

Route-Based VPNs	Policy-Based VPNs
With route-based VPNs, a policy does not specifically reference a VPN tunnel.	With policy-based VPN tunnels, a tunnel is treated as an object that, together with source, destination, application, and action, constitutes a tunnel policy that permits VPN traffic.

Table 6: Differences Between Route-Based VPNs and Policy-Based VPNs *(Continued)*

Route-Based VPNs	Policy-Based VPNs
The policy references a destination address.	In a policy-based VPN configuration, a tunnel policy specifically references a VPN tunnel by name.
The number of route-based VPN tunnels that you create is limited by the number of route entries or the number of st0 interfaces that the device supports, whichever number is lower.	The number of policy-based VPN tunnels that you can create is limited by the number of policies that the device supports.
Route-based VPN tunnel configuration is a good choice when you want to conserve tunnel resources while setting granular restrictions on VPN traffic.	With a policy-based VPN, although you can create numerous tunnel policies referencing the same VPN tunnel, each tunnel policy pair creates an individual IPsec security association (SA) with the remote peer. Each SA counts as an individual VPN tunnel.
With a route-based approach to VPNs, the regulation of traffic is not coupled to the means of its delivery. You can configure dozens of policies to regulate traffic flowing through a single VPN tunnel between two sites, and only one IPsec SA is at work. Also, a route-based VPN configuration allows you to create policies referencing a destination reached through a VPN tunnel in which the action is deny.	In a policy-based VPN configuration, the action must be permit and must include a tunnel.
Route-based VPNs support the exchange of dynamic routing information through VPN tunnels. You can enable an instance of a dynamic routing protocol, such as OSPF, on an st0 interface that is bound to a VPN tunnel.	The exchange of dynamic routing information is not supported in policy-based VPNs.
Route-based configurations are used for hub-and-spoke topologies.	Policy-based VPNs cannot be used for hub-and-spoke topologies.
With route-based VPNs, a policy does not specifically reference a VPN tunnel.	When a tunnel does not connect large networks running dynamic routing protocols and you do not need to conserve tunnels or define various policies to filter traffic through the tunnel, a policy-based tunnel is the best choice.

Table 6: Differences Between Route-Based VPNs and Policy-Based VPNs *(Continued)*

Route-Based VPNs	Policy-Based VPNs
Route-based VPNs do not support remote-access (dial-up) VPN configurations.	Policy-based VPN tunnels are required for remote-access (dial-up) VPN configurations.
Route-based VPNs might not work correctly with some third-party vendors.	Policy-based VPNs might be required if the third party requires separate SAs for each remote subnet.
When the security device does a route lookup to find the interface through which it must send traffic to reach an address, it finds a route via a secure tunnel interface (`st0`) , which is bound to a specific VPN tunnel. With a route-based VPN tunnel, you can consider a tunnel as a means for delivering traffic, and can consider the policy as a method for either permitting or denying the delivery of that traffic.	With a policy-based VPN tunnel, you can consider a tunnel as an element in the construction of a policy.
Route-based VPNs support NAT for st0 interfaces.	Policy-based VPNs cannot be used if NAT is required for tunneled traffic.

Proxy ID is supported for both route-based and policy-based VPNs. Route-based tunnels also offer the usage of multiple traffic selectors also known as multi-proxy ID. A traffic selector is an agreement between IKE peers to permit traffic through a tunnel, if the traffic matches a specified pair of local and remote IP address prefix, source port range, destination port range, and protocol. You define a traffic selector within a specific route-based VPN, which can result in multiple Phase 2 IPsec SAs. Only traffic that conforms to a traffic selector is permitted through an SA. The traffic selector is commonly required when remote gateway devices are non-Juniper Networks devices.

Policy-based VPNs are only supported on SRX5400, SRX5600, and SRX5800 line. Platform support depends on the Junos OS release in your installation.

SEE ALSO

Example: Configuring a Route-Based VPN | 395

Example: Configuring a Policy-Based VPN | 268

Comparison of Policy-Based VPNs and Route-Based VPNs

Table 7 on page 169 summarizes the differences between policy-based VPNs and route-based VPNs.

Table 7: Comparison Between Policy-Based VPNs and Route-Based VPNs

Policy-Based VPNs	Route-Based VPNs
In policy-based VPNs, a tunnel is treated as an object that, together with source, destination, application, and action, constitutes a tunnel policy that permits VPN traffic.	In route-based VPNs, a policy does not specifically reference a VPN tunnel.
A tunnel policy specifically references a VPN tunnel by name.	A route determines which traffic is sent through the tunnel based on a destination IP address.
The number of policy-based VPN tunnels that you can create is limited by the number of tunnels that the device supports.	The number of route-based VPN tunnels that you create is limited by the number of st0 interfaces (for point-to-point VPNs) or the number of tunnels that the device supports, whichever is lower.
With a policy-based VPN, although you can create numerous tunnel policies referencing the same VPN tunnel, each tunnel policy pair creates an individual IPsec SA with the remote peer. Each SA counts as an individual VPN tunnel.	Because the route, not the policy, determines which traffic goes through the tunnel, multiple policies can be supported with a single SA or VPN.
In a policy-based VPN, the action must be permit and must include a tunnel.	In a route-based VPN, the regulation of traffic is not coupled to the means of its delivery.
The exchange of dynamic routing information is not supported in policy-based VPNs.	Route-based VPNs support the exchange of dynamic routing information through VPN tunnels. You can enable an instance of a dynamic routing protocol, such as OSPF, on an st0 interface that is bound to a VPN tunnel.
If you need more granularity than a route can provide to specify the traffic sent to a tunnel, using a policy-based VPN with security policies is the best choice.	Route-based VPNs uses routes to specify the traffic sent to a tunnel; a policy does not specifically reference a VPN tunnel.

Table 7: Comparison Between Policy-Based VPNs and Route-Based VPNs *(Continued)*

Policy-Based VPNs	Route-Based VPNs
With a policy-based VPN tunnel, you can consider a tunnel as an element in the construction of a policy.	When the security device does a route lookup to find the interface through which it must send traffic to reach an address, it finds a route through a secure tunnel (st0) interface. With a route-based VPN tunnel, you can consider a tunnel as a means for delivering traffic, and can consider the policy as a method for either permitting or denying the delivery of that traffic.

Understanding IKE and IPsec Packet Processing

IN THIS SECTION

An IPsec VPN tunnel consists of tunnel setup and applied security. During tunnel setup, the peers establish security associations (SAs), which define the parameters for securing traffic between themselves. (See "IPsec Overview" on page 20.) After the tunnel is established, IPsec protects the traffic sent between the two tunnel endpoints by applying the security parameters defined by the SAs during tunnel setup. Within the Junos OS implementation, IPsec is applied in tunnel mode, which supports the Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols.

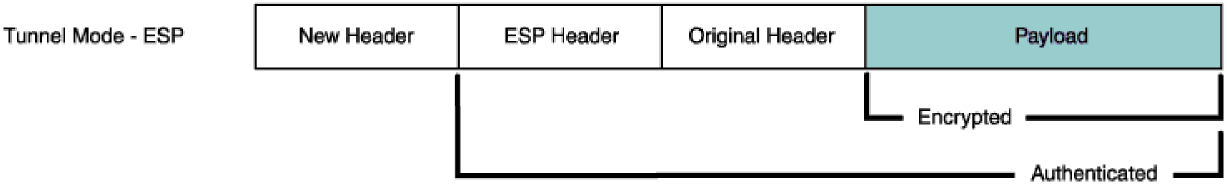
This topic includes the following sections:

Packet Processing in Tunnel Mode

IPsec operates in one of two modes—transport or tunnel. When both ends of the tunnel are hosts, you can use either mode. When at least one of the endpoints of a tunnel is a security gateway, such as a Junos OS router or firewall, you must use tunnel mode. Juniper Networks devices always operate in tunnel mode for IPsec tunnels.

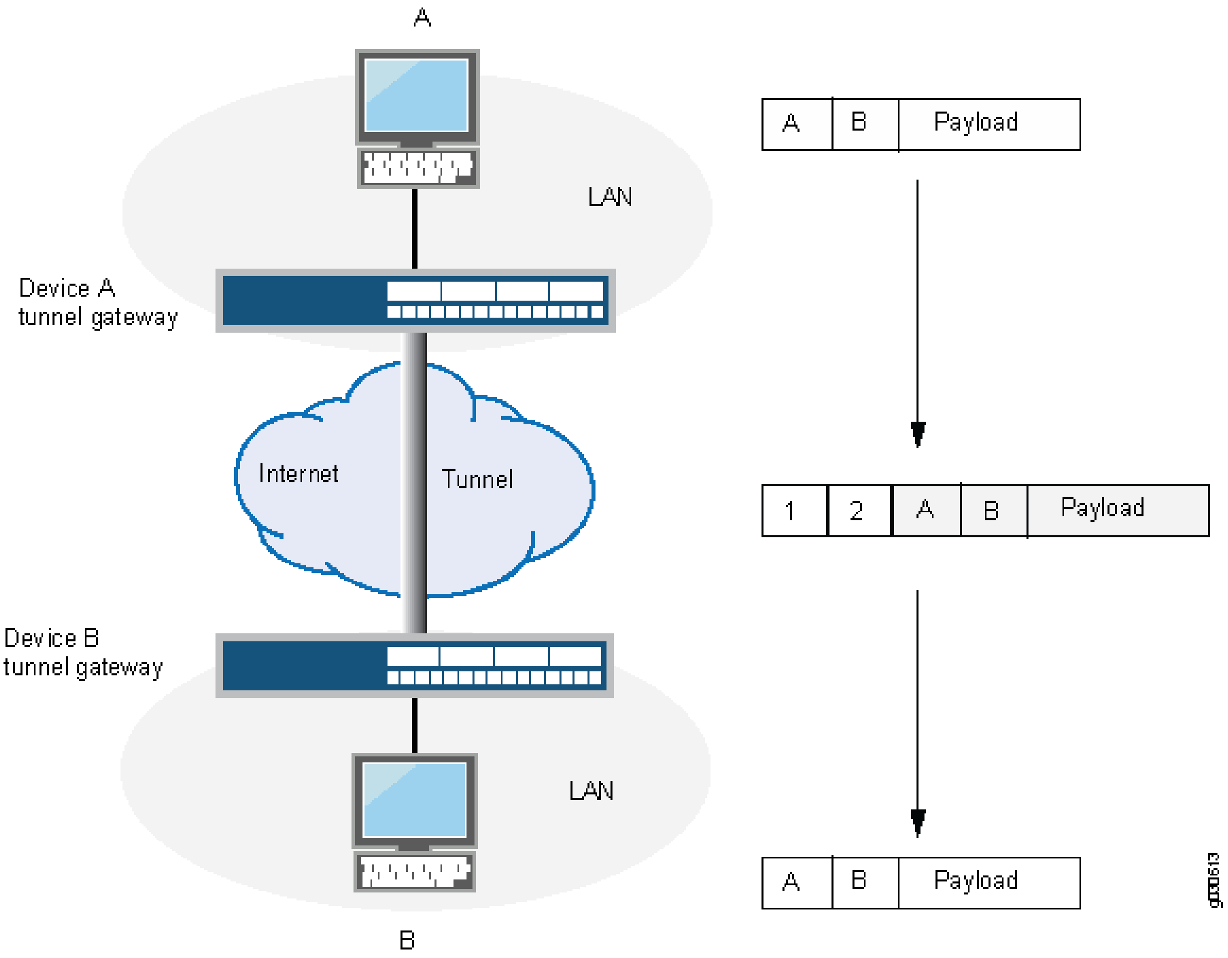
In tunnel mode, the entire original IP packet—payload and header—is encapsulated within another IP payload, and a new header is appended to it, as shown in Figure 21 on page 171. The entire original packet can be encrypted, authenticated, or both. With the Authentication Header (AH) protocol, the AH and new headers are also authenticated. With the Encapsulating Security Payload (ESP) protocol, the ESP header can also be authenticated.

Figure 21: Tunnel Mode



In a site-to-site VPN, the source and destination addresses used in the new header are the IP addresses of the outgoing interface. See Figure 22 on page 172.

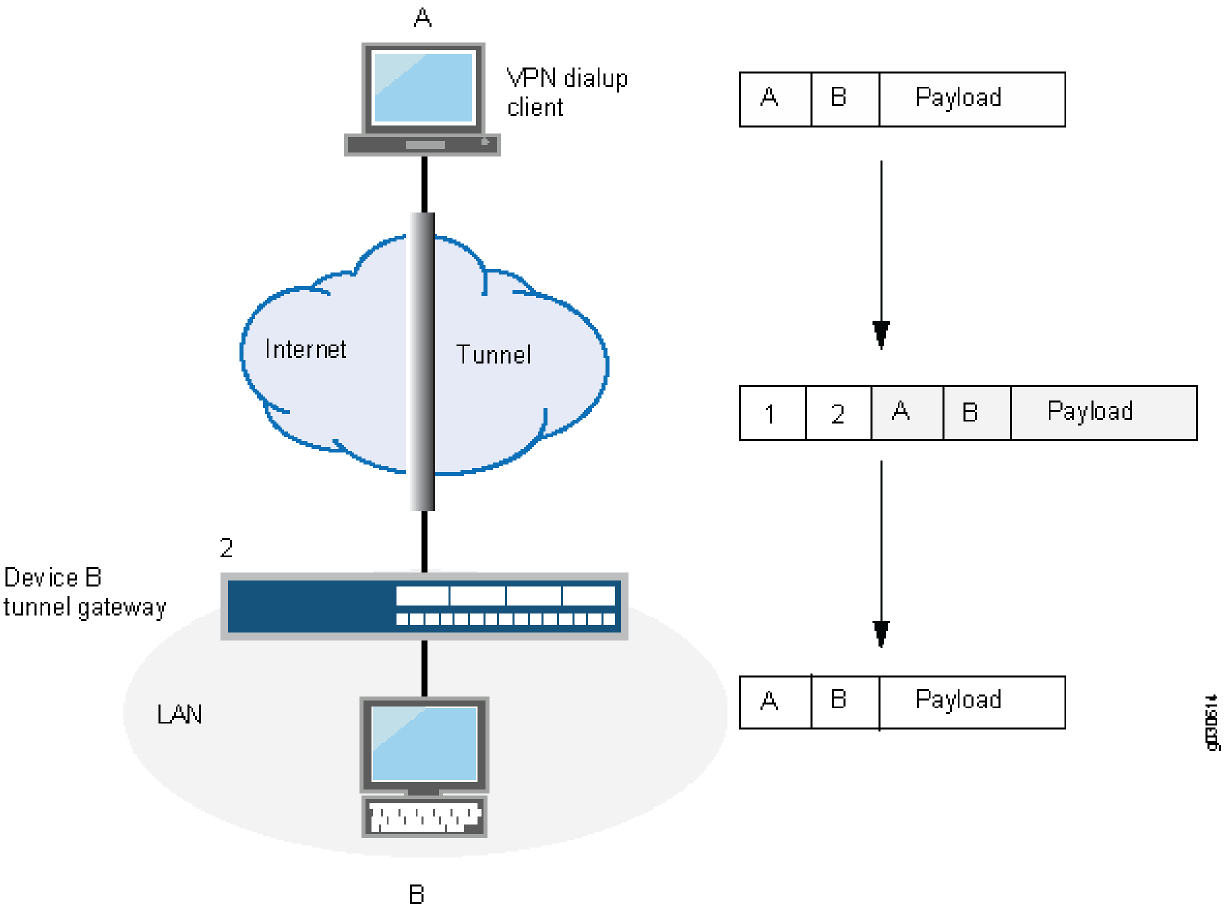
Figure 22: Site-to-Site VPN in Tunnel Mode



In a dial-up VPN, there is no tunnel gateway on the VPN dial-up client end of the tunnel; the tunnel extends directly to the client itself (see Figure 23 on page 173). In this case, on packets sent from the dial-up client, both the new header and the encapsulated original header have the same IP address: that of the client's computer.

Some VPN clients, such as the *dynamic VPN* client and Netscreen-Remote, use a virtual inner IP address (also called a "sticky address"). Netscreen-Remote enables you to define the virtual IP address. The dynamic VPN client uses the virtual IP address assigned during the XAuth configuration exchange. In such cases, the virtual inner IP address is the source IP address in the original packet header of traffic originating from the client, and the IP address that the ISP dynamically assigns the dial-up client is the source IP address in the outer header. Starting in Junos OS Release 21.4R1 dynamic VPN is not supported on SRX300, SRX320, SRX340, SRX345, SRX380, and SRX550 HM devices.

Figure 23: Dial-Up VPN in Tunnel Mode



SEE ALSO

Example: Configuring a Policy-Based VPN | 268

Route-Based IPsec VPNs | 394

Distribution of IKE and IPsec Sessions Across SPUs

In the SRX5400, SRX5600, and SRX5800 devices, IKE provides tunnel management for IPsec and authenticates end entities. IKE performs a Diffie-Hellman (DH) key exchange to generate an IPsec tunnel between network devices. The IPsec tunnels generated by IKE are used to encrypt, decrypt, and authenticate user traffic between the network devices at the IP layer.

The VPN is created by distributing the IKE and IPsec workload among the multiple Services Processing Units (SPUs) of the platform. For site-to-site tunnels, the least-loaded SPU is chosen as the anchor SPU.

If multiple SPUs have the same smallest load, any of them can be chosen as an anchor SPU. Here, load corresponds to the number of site-to-site gateways or manual VPN tunnels anchored on an SPU. For dynamic tunnels, the newly established dynamic tunnels employ a round-robin algorithm to select the SPU.

In IPsec, the workload is distributed by the same algorithm that distributes the IKE. The Phase 2 SA for a given VPN tunnel termination points pair is exclusively owned by a particular SPU, and all IPsec packets belonging to this Phase 2 SA are forwarded to the anchoring SPU of that SA for IPsec processing.

Multiple IPsec sessions (Phase 2 SA) can operate over one or more IKE sessions. The SPU that is selected for anchoring the IPsec session is based on the SPU that is anchoring the underlying IKE session. Therefore, all IPsec sessions that run over a single IKE gateway are serviced by the same SPU and are not load-balanced across several SPUs.

Table 8 on page 174 shows an example of an SRX5000 line with three SPUs running seven IPsec tunnels over three IKE gateways.

Table 8: Distribution of IKE and IPsec Sessions Across SPUs

SPU	IKE Gateway	IPsec Tunnel
SPU0	IKE-1	IPsec-1
		IPsec-2
		IPsec-3
SPU1	IKE-2	IPsec-4
		IPsec-5
		IPsec-6
SPU2	IKE-3	IPsec-7

The three SPUs have an equal load of one IKE gateway each. If a new IKE gateway is created, SPU0, SPU1, or SPU2 could be selected to anchor the IKE gateway and its IPsec sessions.

Setting up and tearing down existing IPsec tunnels does not affect the underlying IKE session or existing IPsec tunnels.

Use the following `show` command to view the current tunnel count per SPU: `show security ike tunnel-map`.

Use the `summary` option of the command to view the anchor points of each gateway: `show security ike tunnel-map summary`.

VPN Support for Inserting Services Processing Cards

SRX5400, SRX5600, and SRX5800 devices have a chassis-based distributed processor architecture. The flow processing power is shared and is based on the number of Services Processing Cards (SPCs). You can scale the processing power of the device by installing new SPCs.

In an SRX5400, SRX5600, or SRX5800 chassis cluster, you can insert new SPCs on the devices without affecting or disrupting the traffic on the existing IKE or IPsec VPN tunnels. When you insert a new SPC in each chassis of the cluster, the existing tunnels are not affected and traffic continues to flow without disruption.

Starting in Junos OS Release 19.4R1, on all SRX5000 line chassis cluster, you can insert a new SRX5K-SPC3 (SPC3) or SRX5K-SPC-4-15-320 (SPC2) card to an existing chassis containing SPC3 card. You can only insert the cards in a higher slot than the existing SPC3 card on the chassis. You must reboot the node after the inserting SPC3 to activate the card. After the node reboot is complete, IPsec tunnels are distributed to the cards.

However, existing tunnels cannot use the processing power of the Service Processing Units (SPUs) in the new SPCs. A new SPU can anchor newly established site-to-site and dynamic tunnels. Newly configured tunnels are not, however, guaranteed to be anchored on a new SPU.

Site-to-site tunnels are anchored on different SPUs based on a load-balancing algorithm. The load-balancing algorithm is dependent on number flow threads each SPU is using. Tunnels belonging to the same local and remote gateway IP addresses are anchored on the same SPU on different flow RT threads used by the SPU. The SPU with the smallest load is chosen as the anchor SPU. Each SPU maintains number of flow RT threads that are hosted in that particular SPU. The number of flow RT threads hosted on each SPU vary based on the type of SPU.

Tunnel load factor = Number of tunnels anchored on the SPU / Total number of flow RT threads used by the SPU.

Dynamic tunnels are anchored on different SPUs based on a round-robin algorithm. Newly configured dynamic tunnels are not guaranteed to be anchored on the new SPC.

Starting in Junos OS Release 18.2R2 and 18.4R1, all the existing IPsec VPN features that are currently supported on SRX5K-SPC3 (SPC3) only will be supported on SRX5400, SRX5600, and SRX5800 devices when SRX5K-SPC-4-15-320 (SPC2) and SPC3 cards are installed and operating on the device in a chassis cluster mode or in a standalone mode.

When both SPC2 and SPC3 cards are installed, you can verify the tunnel mapping on different SPUs using the `show security ipsec tunnel-distribution` command.

Use the command `show security ike tunnel-map` to view the tunnel mapping on different SPUs with only SPC2 card inserted. The command `show security ike tunnel-map` is not valid in an environment where SPC2 and SPC3 cards are installed.

Inserting SPC3 Card: Guidelines and Limitations:

- In a chassis cluster, if one of the nodes has 1 SPC3 card and the other node has 2 SPC3 cards, the failover to the node that has 1 SPC3 card is not supported.
- You must insert the SPC3 or SPC2 in an existing chassis in a higher slot than a current SPC3 present in a lower slot.
- For SPC3 ISHU to work, you must insert the new SPC3 card into the higher slot number.
- On SRX5800 chassis cluster, you must not insert the SPC3 card in the highest slot (slot no. 11) due to the power and heat distribution limit.
- We do not support SPC3 hot removal.

Table 9 on page 176 summarizes the SRX5000 line with SPC2 or SPC3 card that supports `kmd` or `iked` process:

Table 9: `kmd`/`iked` Process Support on SRX5000 Line

SRX5000 Line	Support for `kmd` or `iked` Process
SRX5000 line with only SPC2 card installed	Supports `kmd` process.
SRX5000 line with only SPC3 card installed	Supports `iked` process.
SRX5000 line with both SPC2 and SPC3 card installed	Supports `iked` process.

SEE ALSO

show security ike tunnel-map | 1880

Cryptographic acceleration support on SRX5K-SPC3 Card, SRX mid-range platforms and vSRX Virtual Firewall

SRX5000 line with SRX5K-SPC3 card (Services Processing Card), SRX mid-range platforms (SRX4100, SRX4200, SRX1500 and SRX4600 Series Firewalls) and vSRX Virtual Firewall requires `junos-ike` package as the control plane software to install and enable IPsec VPN features.

- On SRX5000 line with RE3, by default, `junos-ike` package is installed in Junos OS Releases 20.1R2, 20.2R2, 20.3R2, 20.4R1, and later. As a result, **iked** and **ikemd** process runs on the routing engine by default instead of IPsec key management daemon (kmd). The SRX5000 line with SRX5K-SPC3 offloads the cryptographic operations to the hardware cryptographic engine.

- The SRX mid-range platforms covering SRX1500, SRX4100, SRX4200 and SRX4600 Series Firewalls, offloads the DH, RSA and ECDSA cryptographic operations to the hardware cryptographic engine with devices running `junos-ike` software. This feature is available from Junos OS Release 23.2R1 with devices having `junos-ike` package installed. The devices that continue to run legacy **iked** software (kmd process) do not support this feature.

- On vSRX Virtual Firewall, the data plane CPU thread offloads the DH, RSA and ECDSA operations. Hardware acceleration is not available on these devices. This feature is available from Junos OS Release 23.2R1 with `junos-ike` package installed on the device.

The Table 10 on page 177 describes the hardware acceleration support for various ciphers:

Table 10: Cryptographic Acceleration Support

Ciphers	SRX1500		SRX4100/ SRX4200		SRX4600		SRX5K - SPC3	vSRX3.0	
	KMD	IKED	KMD	IKED	KMD	IKED	IKED	KMD	IKED
DH (Groups 1, 2, 5, 14)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DH (Groups 19, 20)	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes
DH (Groups 15, 16)	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes
DH Group 21	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes

Table 10: Cryptographic Acceleration Support *(Continued)*

Ciphers	SRX1500		SRX4100/ SRX4200		SRX4600		SRX5K - SPC3	vSRX3.0	
DH Group 24	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
RSA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ECDSA (256, 384, 521)	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes

To install the Junos IKE package on your SRX Series Firewall, use the following command:

```
user@host> request system software add optional://junos-ike.tgz
Verified junos-ike signed by PackageProductionECP256_2022 method ECDSA256+SHA256
Rebuilding schema and Activating configuration...
mgd: commit complete
Restarting MGD ...

WARNING: cli has been replaced by an updated version:
CLI release 20220208.163814_builder.r1239105 built by builder on 2022-02-08 17:07:55 UTC
Restart cli using the new version ? [yes,no] (yes)
```

To use `kmd` process in order to enable IPsec VPN features on SRX5000 line without an SPC3 card, you must run the `request system software delete junos-ike` command. After running the command, reboot the device.

To check the installed `junos-ike` package, use the following command:

```
user@host> show version | grep ike

JUNOS ike [20190617.180318_builder_junos_182_x41]
JUNOS ike [20190617.180318_builder_junos_182_x41]

{primary:node0}
```

SEE ALSO

- show security ipsec security-associations | 1899
- show security ipsec tunnel-distribution | 1949

IPsec VPN Feature Support on SRX5000 Line with SRX5K-SPC3 and vSRX Virtual Firewall Instances with New Package

IN THIS SECTION

This topic provides you a summary of IPsec VPN features and configurations that are not supported of SRX5000 line with SRX5K-SPC3 and on vSRX Virtual Firewall instances.

IPsec VPN feature is supported by two processes, **iked** and **ikemd** on SRX5K-SPC3 and vSRX Virtual Firewall instances. A single instance of **iked** and **ikemd** will run on the Routing Engine at a time.

By default, `Junos-ike` package is installed in Junos OS Releases 20.1R2, 20.2R2, 20.3R2, 20.4R1, and later for SRX5000 line with RE3, and both the **iked** and **ikemd** process runs on the routing engine.

To restart **ikemd** process in the Routine Engine use the `restart ike-config-management` command.

To restart **iked** process in the Routing Engine use the `restart ike-key-management` command.

If you want to use `kmd` process to enable IPsec VPN features on SRX5000 line without an SRX5K-SPC3 card, you must run the `request system software delete junos-ike` command. After running the command, you must reboot the device.

IPsec VPN Features Not Supported

To determine if a feature is supported by a specific platform or Junos OS release, refer Feature Explorer.

Table 11 on page 180 summarizes the non-supported IPsec VPN features on SRX Series Firewalls and vSRX Virtual Firewall running iked process:

Table 11: IPsec VPN Features Not Supported on SRX Series Firewalls and vSRX Virtual Firewall Instances

Features	Support on SRX5000 line with SRX5K-SPC3 and vSRX Virtual Firewall Instances
Auto Discovery VPN (ADVPN).	No
AutoVPN Protocol Independent Multicast (PIM) point-to-multipoint mode.	No
Configuring forwarding class on IPsec VPNs.	No
Dead peer detection (DPD) gateway failover.	DPD gateway failover is not supported on vSRX Virtual Firewall.
Group VPN.	No
Lifetime of IKE SA, in kilobytes.	No
Packet size configuration for IPsec datapath verification.	No
Policy-based IPsec VPN.	No
VPN monitoring.	No

Routing Protocols Support on IPsec VPN Tunnels

We support routing protocols like, OSPF, BGP, PIM, RIP, and BFD to run on IPsec tunnels on SRX Series Firewalls and MX Series routers running `kmd` or `iked` process. The protocol support varies based on the IP addressing scheme or the type of the st0 interface, point-to-point (P2P) or point-to-multipoint (P2MP).

Table 12 on page 181 summarizes OSPF protocol support on SRX Series Firewalls and MX routers.

Table 12: OSPF Protocol Support on IPsec VPN Tunnels

OSPF				
Devices	P2P		P2MP	
	IPv4	IPv6	IPv4	IPv6
SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550, SRX550 HM, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2	Yes	No	Yes	No
SRX5K-SPC3	Yes	No	Yes	No
SRX5K in mixed-mode (SPC3 + SPC2)	Yes	No	Yes	No
vSRX Virtual Firewall 3.0	Yes	No	Yes	No
MX-SPC3	Yes	No	No	No

Table 13 on page 182 summarizes OSPFv3 protocol support on SRX Series Firewalls and MX routers.

Table 13: OSPFv3 Protocol Support on IPsec VPN Tunnels

OSPFv3				
Devices	P2P		P2MP	
	IPv4	IPv6	IPv4	IPv6
SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550, SRX550 HM, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2	No	Yes	No	Yes
SRX5K-SPC3	No	Yes	No	Yes
SRX5K in mixed-mode (SPC3 + SPC2)	No	Yes	No	Yes
vSRX Virtual Firewall 3.0	No	Yes	No	Yes
MX-SPC3	No	Yes	No	No

Table 14 on page 183 summarizes BGP protocol support on SRX Series Firewalls and MX routers.

Table 14: BGP Protocol Support on IPsec VPN Tunnels

BGP				
Devices	P2P		P2MP	
	IPv4	IPv6	IPv4	IPv6
SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550, SRX550 HM, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2	Yes	Yes	Yes	Yes
SRX5K-SPC3	Yes	Yes	Yes	Yes
SRX5K in mixed-mode (SPC3 + SPC2)	Yes	Yes	Yes	Yes
vSRX Virtual Firewall 3.0	Yes	Yes	Yes	Yes
MX-SPC3	Yes	Yes	No	No

Table 15 on page 184 summarizes PIM protocol support on SRX Series Firewalls and MX routers.

Table 15: PIM Protocol Support on IPsec VPN Tunnels

PIM				
Devices	P2P		P2MP	
	IPv4	IPv6	IPv4	IPv6
SRX100, SRX110, SRX210, SRX220, SRX240, SRX550, SRX550 HM, SRX650, SRX1400, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2	Yes	No	No	No
SRX300, SRX320, SRX340, SRX345, SRX380, and SRX1500	Yes	No	Yes	No
SRX5K-SPC3	Yes	No	No	No
SRX5K in mixed-mode (SPC3 + SPC2)	Yes	No	No	No
vSRX Virtual Firewall	Yes	No	Yes **NOTE**: Multithread is not supported.	No
MX-SPC3	Yes	No	No	No

Table 16 on page 185 summarizes RIP protocol support on SRX Series Firewalls and MX routers.

Table 16: RIP Protocol Support on IPsec VPN Tunnels

RIP				
Devices	P2P		P2MP	
	IPv4	IPv6	IPv4	IPv6
SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550, SRX550 HM, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2	Yes	Yes	No	No
SRX5K-SPC3	Yes	Yes	No	No
SRX5K in mixed-mode (SPC3 + SPC2)	Yes	Yes	No	No
vSRX Virtual Firewall 3.0	Yes	Yes	No	No
MX-SPC3	Yes	Yes	No	No

Table 17 on page 186 summarizes BFP protocol support on SRX Series Firewalls and MX routers.

Table 17: BFD Protocol Support on IPsec VPN Tunnels

BFD				
Devices	P2P		P2MP	
	IPv4	IPv6	IPv4	IPv6
SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX380, SRX550, SRX550 HM, SRX650, SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, and SRX5K-SPC2	Yes	Yes	Yes	Yes
SRX5K-SPC3	Yes	Yes	Yes	Yes
SRX5K in mixed-mode (SPC3 + SPC2)	Yes	Yes	Yes	Yes
vSRX Virtual Firewall 3.0	Yes	Yes	Yes	Yes
MX-SPC3	Yes	Yes	No	No

Anti-Replay Window

On SRX Series Firewalls, `anti-replay-window` is enabled by default with a window size value of 64.

On the SRX Series 5000 line with SPC3 cards installed, you can configure the `anti-replay-window` size in the range of 64 to 8192 (power of 2). To configure the window size, use the new `anti-replay-window-size` option. An incoming packet is validated for replay attack based on the anti-replay-window-size that is configured.

You can configure replay-window-size at two different levels:

- **Global level**—Configured at the [edit security ipsec] hierarchy level.

  For example:

  ```
  [edit security ipsec]
   user@host# set anti-replay-window-size <64..8192>;
  ```

- **VPN object**—Configured at the [edit security ipsec vpn vpn-name ike] hierarchy level.

  For example:

  ```
  [edit security ipsec vpn vpn-name ike]
  user@host# set anti-replay-window-size <64..8192>;
  ```

If anti-replay is configured at both levels, the window size configured for a VPN object level takes precedence over the window size configured at the global level. If anti-replay is not configured, the window size is 64 by default.

To disable the anti-replay window option on a VPN object, use the set no-anti-replay command at the [edit security ipsec vpn vpn-name ike] hierarchy level. You cannot disable anti-replay at the global level.

You cannot configure both anti-replay-window-size and no-anti-replay on a VPN object.

SEE ALSO

anti-replay-window-size | **1446**

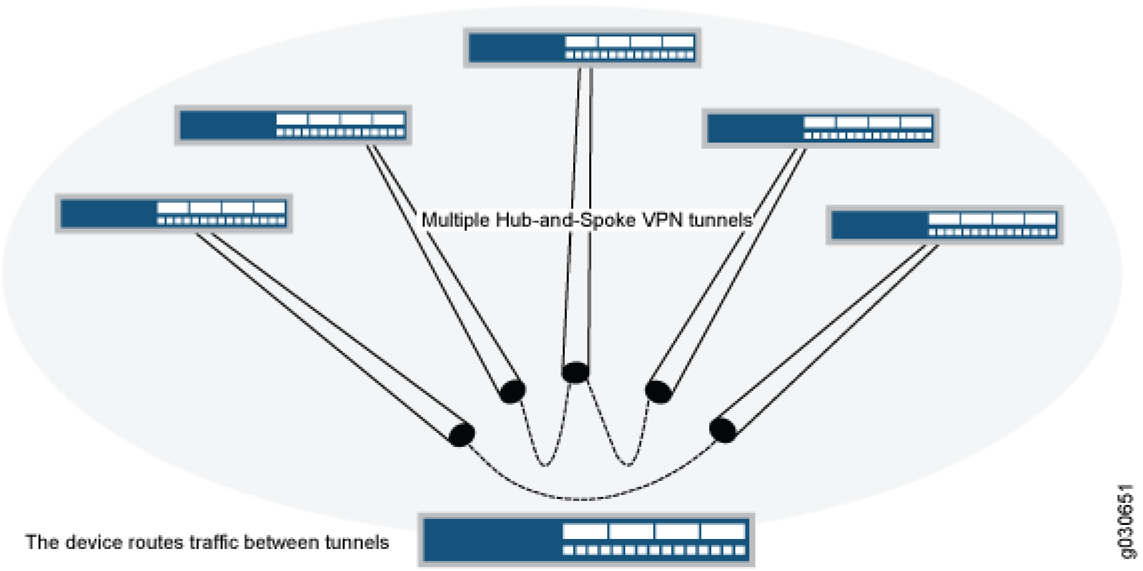
Understanding Hub-and-Spoke VPNs

If you create two VPN tunnels that terminate at a device, you can set up a pair of routes so that the device directs traffic exiting one tunnel to the other tunnel. You also need to create a policy to permit the traffic to pass from one tunnel to the other. Such an arrangement is known as *hub-and-spoke VPN*. (See Figure 24 on page 188.)

You can also configure multiple VPNs and route traffic between any two tunnels.

SRX Series Firewalls support only the route-based hub-and-spoke feature.

Figure 24: Multiple Tunnels in a Hub-and-Spoke VPN Configuration


Multiple Hub-and-Spoke VPN tunnels
The device routes traffic between tunnels
g030651

SEE ALSO

Example: Configuring a Hub-and-Spoke VPN | 209

Release History Table

Release	Description
23.2R1	Cryptographic acceleration support for SRX mid-range platforms (SRX1500, SRX4100, SRX4200, SRX4600 Series Firewalls) and vSRX Virtual Firewall is added.
20.1R2	By default, `junos-ike` package is installed in Junos OS Releases 20.1R2, 20.2R2, 20.3R2, 20.4R1, and later for SRX5000 line with RE3. As a result **iked** and **ikemd** process runs on the routing engine by default instead of IPsec key management daemon (kmd).

RELATED DOCUMENTATION

Route-Based IPsec VPNs | 394

Policy-Based IPsec VPNs | 267

5

CHAPTER

VPN Configuration Overview

IPsec VPN Configuration Overview

IN THIS SECTION

A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. An IPsec tunnel is created between two participant devices to secure VPN communication.

IPsec VPN with Autokey IKE Configuration Overview

IPsec VPN negotiation occurs in two phases. In Phase 1, participants establish a secure channel in which to negotiate the IPsec security association (SA). In Phase 2, participants negotiate the IPsec SA for authenticating traffic that will flow through the tunnel.

This overview describes the basic steps to configure a route-based or policy-based IPsec VPN using autokey IKE (preshared keys or certificates).

To configure a route-based or policy-based IPsec VPN using autokey IKE:

1. Configure interfaces, security zones, and address book information.

   (For route-based VPNs) Configure a secure tunnel st0.x interface. Configure routing on the device.
2. Configure Phase 1 of the IPsec VPN tunnel.

   a. (Optional) Configure a custom IKE Phase 1 proposal. This step is optional, as you can use a predefined IKE Phase 1 proposal set (Standard, Compatible, or Basic).

   b. Configure an IKE policy that references either your custom IKE Phase 1 proposal or a predefined IKE Phase 1 proposal set. Specify autokey IKE preshared key or certificate information. Specify the mode (main or aggressive) for the Phase 1 exchanges.

   c. Configure an IKE gateway that references the IKE policy. Specify the IKE IDs for the local and remote devices. If the IP address of the remote gateway is not known, specify how the remote gateway is to be identified.

3. Configure Phase 2 of the IPsec VPN tunnel.

   a. (Optional) Configure a custom IPsec Phase 2 proposal. This step is optional, as you can use a predefined IPsec Phase 2 proposal set (Standard, Compatible, or Basic).

   b. Configure an IPsec policy that references either your custom IPsec Phase 2 proposal or a predefined IPsec Phase 2 proposal set. Specify perfect forward secrecy (PFS) keys.

   c. Configure an IPsec VPN tunnel that references both the IKE gateway and the IPsec policy. Specify the proxy IDs to be used in Phase 2 negotiations.

      (For route-based VPNs) Bind the secure tunnel interface st0.x to the IPsec VPN tunnel.

4. Configure a security policy to permit traffic from the source zone to the destination zone.

   (For policy-based VPNs) Specify the security policy action `tunnel ipsec-vpn` with the name of the IPsec VPN tunnel that you configured.

5. Update your global VPN settings.

SEE ALSO

Understanding Route-Based IPsec VPNs | **394**

Understanding Policy-Based IPsec VPNs | **267**

Recommended Configuration Options for Site-to-Site VPN with Static IP Addresses

Table 18 on page 192 lists the configuration options for a generic site-to-site VPN between two security devices with static IP addresses. The VPN can be either route-based or policy-based.

Table 18: Recommended Configuration for Site-to-Site VPN with Static IP Addresses

Configuration Option	Comment
IKE configuration options:	
Main mode	Used when peers have static IP addresses.
RSA or DSA certificates	RSA or DSA certificates can be used on the local device. Specify the type of certificate (PKCS7 or X.509) on the peer.
Diffie-Hellman (DH) group 14	DH group 14 provides more security than DH groups 1, 2, or 5.
Advanced Encryption Standard (AES) encryption	AES is cryptographically stronger than Data Encryption Standard (DES) and Triple DES (3DES) when key lengths are equal. Approved encryption algorithm for Federal Information Processing Standards (FIPS) and Common Criteria EAL4 standards.
Secure Hash Algorithm 256 (SHA-256) authentication	SHA-256 provides more cryptographic security than SHA-1 or Message Digest 5 (MD5) .
IPsec configuration options:	
Perfect Forward Secrecy (PFS) DH group 14	PFS DH group 14 provides increased security because the peers perform a second DH exchange to produce the key used for IPsec encryption and decryption.
Encapsulating Security Payload (ESP) protocol	ESP provides both confidentiality through encryption and encapsulation of the original IP packet and integrity through authentication.
AES encryption	AES is cryptographically stronger than DES and 3DES when key lengths are equal. Approved encryption algorithm for FIPS and Common Criteria EAL4 standards.
SHA-256 authentication	SHA-256 provides more cryptographic security than SHA-1 or MD5.

Table 18: Recommended Configuration for Site-to-Site VPN with Static IP Addresses *(Continued)*

Configuration Option	Comment
Anti-replay protection	Enabled by default. Disabling this feature might resolve compatibility issues with third-party peers.

SEE ALSO

IPsec Overview | 20

Recommended Configuration Options for Site-to-Site or Dialup VPNs with Dynamic IP Addresses

Table 19 on page 193 lists the configuration options for a generic site-to-site or dialup VPN, where the peer devices have dynamic IP addresses.

Table 19: Recommended Configuration for Site-to-Site or Dialup VPNs with Dynamic IP Addresses

Configuration Option	Comment
IKE configuration options:	
Main mode	Used with certificates.
2048-bit certificates	RSA or DSA certificates can be used. Specify the certificate to be used on the local device. Specify the type of certificate (PKCS7 or X.509) on the peer.
Diffie-Hellman (DH) group 14	DH group 14 provides more security than DH groups 1, 2, or 5.
Advanced Encryption Standard (AES) encryption	AES is cryptographically stronger than Data Encryption Standard (DES) and Triple DES (3DES) when key lengths are equal. Approved encryption algorithm for Federal Information Processing Standards (FIPS) and Common Criteria EAL4 standards.

Table 19: Recommended Configuration for Site-to-Site or Dialup VPNs with Dynamic IP Addresses *(Continued)*

Configuration Option	Comment
Secure Hash Algorithm 256 (SHA-256) authentication	SHA-256 provides more cryptographic security than SHA-1 or Message Digest 5 (MD5).
IPsec configuration options:	
Perfect Forward Secrecy (PFS) DH group 14	PFS DH group 14 provides increased security because the peers perform a second DH exchange to produce the key used for IPsec encryption and decryption.
Encapsulating Security Payload (ESP) protocol	ESP provides both confidentiality through encryption and encapsulation of the original IP packet and integrity through authentication.
AES encryption	AES is cryptographically stronger than DES and 3DES when key lengths are equal. Approved encryption algorithm for FIPS and Common Criteria EAL4 standards.
SHA-256 authentication	SHA-256 provides more cryptographic security than SHA-1 or MD5.
Anti-replay protection	Enabled by default. Disabling this might resolve compatibility issues with third-party peers.

SEE ALSO

IPsec Overview | 20

Understanding IPsec VPNs with Dynamic Endpoints

IN THIS SECTION

Overview

An IPsec VPN peer can have an IP address that is not known to the peer with which it is establishing the VPN connection. For example, a peer can have an IP address dynamically assigned by means of Dynamic Host Configuration Protocol (DHCP). This could be the case with a remote access client in a branch or home office or a mobile device that moves between different physical locations. Or, the peer can be located behind a NAT device that translates the peer's original source IP address into a different address. A VPN peer with an unknown IP address is referred to as a *dynamic endpoint* and a VPN established with a dynamic endpoint is referred to as a *dynamic endpoint VPN*.

On SRX Series Firewalls, IKEv1 or IKEv2 is supported with dynamic endpoint VPNs. Dynamic endpoint VPNs on SRX Series Firewalls support IPv4 traffic on secure tunnels. Starting with Junos OS Release 15.1X49-D80, dynamic endpoint VPNs on SRX Series Firewalls support IPv6 traffic on secure tunnels.

IPv6 traffic is not supported for AutoVPN networks.

The following sections describe items to note when configuring a VPN with a dynamic endpoint.

IKE Identity

On the dynamic endpoint, an IKE identity must be configured for the device to identify itself to its peer. The local identity of the dynamic endpoint is verified on the peer. By default, the SRX Series Firewall expects the IKE identity to be one of the following:

- When certificates are used, a distinguished name (DN) can be used to identify users or an organization.
- A hostname or fully qualified domain name (FQDN) that identifies the endpoint.

- A user fully qualified domain name (UFQDN), also known as *user-at-hostname*. This is a string that follows the e-mail address format.

Aggressive Mode for IKEv1 Policy

When IKEv1 is used with dynamic endpoint VPNs, the IKE policy must be configured for aggressive mode.

IKE Policies and External Interfaces

Starting with Junos OS Release 12.3X48-D40, Junos OS Release 15.1X49-D70, and Junos OS Release 17.3R1, all dynamic endpoint gateways configured on SRX Series Firewalls that use the same external interface can use different IKE policies, but the IKE policies must use the same IKE proposal. This applies to IKEv1 and IKEv2.

NAT

If the dynamic endpoint is behind a NAT device, NAT-T must be configured on the SRX Series Firewall. NAT keepalives might be required to maintain the NAT translation during the connection between the VPN peers. By default, NAT-T is enabled on SRX Series Firewalls and NAT keepalives are sent at 20-second intervals.

Group and Shared IKE IDs

You can configure an individual VPN tunnel for each dynamic endpoint. For IPv4 dynamic endpoint VPNs, you can use the group IKE ID or shared IKE ID features to allow a number of dynamic endpoints to share an IKE gateway configuration.

The group IKE ID allows you to define a common part of a full IKE ID for all dynamic endpoints, such as “example.net.” A user-specific part, such as the username “Bob,” concatenated with the common part forms a full IKE ID (Bob.example.net) that uniquely identifies each user connection.

The shared IKE ID allows dynamic endpoints to share a single IKE ID and preshared key.

SEE ALSO

Example: Configuring NAT-T with Dynamic Endpoint VPN | **678**

Understanding IKE Identity Configuration

IN THIS SECTION

The IKE identification (IKE ID) is used for validation of VPN peer devices during IKE negotiation. The IKE ID received by the SRX Series Firewall from a remote peer can be an IPv4 or IPv6 address, a hostname, a fully qualified domain name (FQDN), a user FQDN (UFQDN), or a distinguished name (DN). The IKE ID sent by the remote peer needs to match what is expected by the SRX Series Firewall. Otherwise, IKE ID validation fails and the VPN is not established.

IKE ID Types

The SRX Series Firewalls support the following types of IKE identities for remote peers:

- An IPv4 or IPv6 address is commonly used with site-to-site VPNs, where the remote peer has a static IP address.

- A hostname is a string that identifies the remote peer system. This can be an FQDN that resolves to an IP address. It can also be a partial FQDN that is used in conjunction with an IKE user type to identify a specific remote user.

  When a hostname is configured instead of an IP address, the committed configuration and subsequent tunnel establishment is based on the currently-resolved IP address. If the remote peer's IP address changes, the configuration is no longer valid.

- A UFQDN is a string that follows the same format as an e-mail address, such as `user@example.com`.

- A DN is a name used with digital certificates to uniquely identify a user. For example, a DN can be "CN=user, DC=example, DC=com." Optionally, you can use the `container` keyword to specify that the order of the fields in a DN and their values exactly match the configured DN, or use the `wildcard` keyword to specify that the values of fields in a DN must match but the order of the fields does not matter.

  Starting in Junos OS Release 19.4R1, you can now configure only one dynamic DN attribute among `container-string` and `wildcard-string` at `[edit security ike gateway` *`gateway_name`* `dynamic distinguished-name]` hierarchy. If you try configuring the second attribute after you configure the first attribute, the first attribute is replaced with the second attribute. Before your upgrade your device, you must remove one of the attributes if you have configured both the attributes.

- An IKE user type can be used with AutoVPN and remote access VPNs when there are multiple remote peers connecting to the same VPN gateway on the SRX Series Firewall. Configure `ike-user-type group-ike-id` to specify a group IKE ID or `ike-user-type shared-ike-id` to specify a shared IKE ID.

Remote IKE IDs and Site-to-Site VPNs

For site-to-site VPNs, the remote peer's IKE ID can be the IP address of the egress network interface card, a loopback address, a hostname, or a manually configured IKE ID, depending on the configuration of the peer device.

By default, SRX Series Firewalls expect the remote peer's IKE ID to be the IP address configured with the `set security ike gateway` *`gateway-name`* `address` configuration. If the remote peer's IKE ID is a different value, you need to configure the `remote-identity` statement at the [`edit security ike gateway` *`gateway-name`*] hierarchy level.

For example, an IKE gateway on the SRX Series Firewalls is configured with the `set security ike gateway remote-gateway address 203.0.113.1` command. However, the IKE ID sent by the remote peer is `host.example.net`. There is a mismatch between what the SRX Series Firewall expects for the remote peer's IKE ID (203.0.113.1) and the actual IKE ID (`host.example.net`) sent by the peer. In this case, IKE ID validation fails. Use the `set security ike gateway remote-gateway remote-identity hostname host.example.net` to match the IKE ID received from the remote peer.

Remote IKE IDs and Dynamic Endpoint VPNs

For dynamic endpoint VPNs, the remote peer's expected IKE ID is configured with the options at the [`edit security ike gateway` *`gateway-name`* `dynamic`] hierarchy level. For AutoVPN, `hostname` combined with `ike-user-type group-ike-id` can be used where there are multiple peers that have a common domain name. If certificates are used for verifying the peer, a DN can be configured.

Local IKE ID of the SRX Series Firewall

By default, the SRX Series Firewall uses the IP address of its external interface to the remote peer as its IKE ID. This IKE ID can be overridden by configuring the `local-identity` statement at the [`edit security ike gateway` *`gateway-name`*] hierarchy level. If you need to configure the `local-identity` statement on an SRX Series Firewall, make sure that the configured IKE ID matches the IKE ID expected by the remote peer.

SEE ALSO

Understanding Spoke Authentication in AutoVPN Deployments | 1027

Configuring Remote IKE IDs for Site-to-Site VPNs

By default, SRX Series Firewalls validate the IKE ID received from the peer with the IP address configured for the IKE gateway. In certain network setups, the IKE ID received from the peer (which can be an IPv4 or IPv6 address, fully qualified domain name [FQDN], distinguished name, or e-mail address) does not match the IKE gateway configured on the SRX Series Firewall. This can lead to a Phase 1 validation failure.

To modify the configuration of the SRX Series Firewall or the peer device for the IKE ID that is used:

- On the SRX Series Firewall, configure the `remote-identity` statement at the [`edit security ike gateway` *`gateway-name`*] hierarchy level to match the IKE ID that is received from the peer. Values can be an IPv4 or IPv6 address, FQDN, distinguished name, or e-mail address.

  If you do not configure `remote-identity`, the device uses the IPv4 or IPv6 address that corresponds to the remote peer by default.

- On the peer device, ensure that the IKE ID is the same as the `remote-identity` configured on the SRX Series Firewall. If the peer device is an SRX Series Firewall, configure the `local-identity` statement at the [`edit security ike gateway` *`gateway-name`*] hierarchy level. Values can be an IPv4 or IPv6 address, FQDN, distinguished name, or e-mail address.

SEE ALSO

Understanding NAT-T | 593
Example: Configuring a Route-Based VPN with the Responder behind a NAT Device | 594
Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device | 631

Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls

OSPFv3 does not have a built-in authentication method and relies on the IP Security (IPsec) suite to provide this functionality. IPsec provides authentication of origin, data integrity, confidentiality, replay protection, and nonrepudiation of source. You can use IPsec to secure specific OSPFv3 interfaces and virtual links and to provide encryption for OSPF packets.

OSPFv3 uses the IP authentication header (AH) and the IP Encapsulating Security Payload (ESP) portions of the IPsec protocol to authenticate routing information between peers. AH can provide connectionless integrity and data origin authentication. It also provides protection against replays. AH authenticates as much of the IP header as possible, as well as the upper-level protocol data. However, some IP header fields might change in transit. Because the value of these fields might not be predictable by the sender, they cannot be protected by AH. ESP can provide encryption and limited traffic flow confidentiality or connectionless integrity, data origin authentication, and an anti-replay service.

IPsec is based on security associations (SAs). An SA is a set of IPsec specifications that are negotiated between devices that are establishing an IPsec relationship. This simplex connection provides security services to the packets carried by the SA. These specifications include preferences for the type of authentication, encryption, and IPsec protocol to be used when establishing the IPsec connection. An SA is used to encrypt and authenticate a particular flow in one direction. Therefore, in normal bidirectional traffic, the flows are secured by a pair of SAs. An SA to be used with OSPFv3 must be configured manually and use transport mode. Static values must be configured on both ends of the SA.

To configure IPsec for OSPF or OSPFv3, first define a manual SA with the `security-association` *`sa-name`* option at the `[edit security ipsec]` hierarchy level. This feature only supports bidirectional manual key SAs in transport mode. Manual SAs require no negotiation between the peers. All values, including the keys, are static and specified in the configuration. Manual SAs statically define the security parameter index (SPI) values, algorithms, and keys to be used and require matching configurations on both endpoints (OSPF or OSPFv3 peers). As a result, each peer must have the same configured options for communication to take place.

The actual choice of encryption and authentication algorithms is left to your IPsec administrator; however, we have the following recommendations:

- Use ESP with null encryption to provide authentication to protocol headers but not to the IPv6 header, extension headers, and options. With null encryption, you are choosing not to provide encryption on protocol headers. This can be useful for troubleshooting and debugging purposes. For more information about null encryption, see RFC 2410, *The NULL Encryption Algorithm and Its Use with IPsec*.

- Use ESP with DES or 3DES for full confidentiality.

- Use AH to provide authentication to protocol headers, immutable fields in IPv6 headers, and extension headers and options.

The configured SA is applied to the OSPF or OSPFv3 configurations as follows:

- For an OSPF or OSPFv3 interface, include the `ipsec-sa` *`name`* statement at the `[edit protocols ospf area` *`area-id`* `interface` *`interface-name`*`]` or `[edit protocols ospf3 area` *`area-id`* `interface` *`interface-name`*`]` hierarchy level. Only one IPsec SA name can be specified for an OSPF or OSPFv3 interface; however, different OSPF/OSPFv3 interfaces can specify the same IPsec SA.

- For an OSPF or OSPFv3 virtual link, include the `ipsec-sa` *`name`* statement at the [`edit protocols ospf area` *`area-id`* `virtual-link neighbor-id` *`router-id`* `transit-area` *`area-id`*] or [`edit protocols ospf3 area` *`area-id`* `virtual-link neighbor-id` *`router-id`* `transit-area` *`area-id`*] hierarchy level. You must configure the same IPsec SA for all virtual links with the same remote endpoint address.

The following restrictions apply to IPsec authentication for OSPF or OSPFv3 on SRX Series Firewalls:

- Manual VPN configurations that are configured at the [`edit security ipsec vpn` *`vpn-name`* `manual`] hierarchy level cannot be applied to OSPF or OSPFv3 interfaces or virtual links to provide IPsec authentication and confidentiality.
- You cannot configure IPsec for OSPF or OSPFv3 authentication if there is an existing IPsec VPN configured on the device with the same local and remote addresses.
- IPsec for OSPF or OSPFv3 authentication is not supported over secure tunnel st0 interfaces.
- Rekeying of manual keys is not supported.
- Dynamic Internet Key Exchange (IKE) SAs are not supported.
- Only IPsec transport mode is supported. In transport mode, only the payload (the data you transfer) of the IP packet is encrypted, authenticated, or both. Tunnel mode is not supported.
- Because only bidirectional manual SAs are supported, all OSPFv3 peers must be configured with the same IPsec SA. You configure a manual bidirectional SA at the [`edit security ipsec`] hierarchy level.
- You must configure the same IPsec SA for all virtual links with the same remote endpoint address.

SEE ALSO

IPsec Overview | **20**

Example: Configuring IPsec Authentication for an OSPF Interface on an SRX Series Firewall

IN THIS SECTION

This example shows how to configure and apply a manual security association (SA) to an OSPF interface.

Requirements

Before you begin:

- Configure the device interfaces.
- Configure the router identifiers for the devices in your OSPF network.
- Control OSPF designated router election.
- Configure a single-area OSPF network.
- Configure a multiarea OSPF network.

Overview

You can use IPsec authentication for both OSPF and OSPFv3. You configure the manual SA separately and apply it to the applicable OSPF configuration. Table 20 on page 202 lists the parameters and values configured for the manual SA in this example.

Table 20: Manual SA for IPsec OSPF Interface Authentication

Parameter	Value
SA name	sa1

Table 20: Manual SA for IPsec OSPF Interface Authentication *(Continued)*

Parameter	Value
Mode	transport
Direction	bidirectional
Protocol	AH
SPI	256
Authentication algorithm Key	hmac-md5-96 (ASCII) 123456789012abc
Encryption algorithm Key	des (ASCII) cba210987654321

Configuration

IN THIS SECTION

Configuring a Manual SA

CLI Quick Configuration

To quickly configure a manual SA to be used for IPsec authentication on an OSPF interface, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
[edit]
set security ipsec security-association sa1
set security ipsec security-association sa1 mode transport
set security ipsec security-association sa1 manual direction bidirectional
set security ipsec security-association sa1 manual direction bidirectional protocol ah
set security ipsec security-association sa1 manual direction bidirectional spi 256
set security ipsec security-association sa1 manual direction bidirectional authentication
algorithm hmac-md5-96 key ascii-text 123456789012abc
set security ipsec security-association sa1 manual direction bidirectional encryption algorithm
des key ascii-text cba210987654321
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure a manual SA:

1. Specify a name for the SA.

   ```
   [edit]
   user@host# edit security ipsec security-association sa1
   ```

2. Specify the mode of the manual SA.

   ```
   [edit security ipsec security-association sa1]
   user@host# set mode transport
   ```

3. Configure the direction of the manual SA.

   ```
   [edit security ipsec security-association sa1]
   user@host# set manual direction bidirectional
   ```

4. Configure the IPsec protocol to use.

```
[edit security ipsec security-association sa1]
user@host# set manual direction bidirectional protocol ah
```

5. Configure the value of the SPI.

```
[edit security ipsec security-association sa1]
user@host# set manual direction bidirectional spi 256
```

6. Configure the authentication algorithm and key.

```
[edit security ipsec security-association sa1]
user@host# set manual direction bidirectional authentication algorithm hmac-md5-96 key ascii-
text 123456789012abc
```

7. Configure the encryption algorithm and key.

```
[edit security ipsec security-association sa1]
user@host# set manual direction bidirectional encryption algorithm des key ascii-text
cba210987654321
```

Results

Confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

After you configure the password, you do not see the password itself. The output displays the encrypted form of the password you configured.

```
[edit]
user@host# show security ipsec
security-association sa1 {
    mode transport;
    manual {
        direction bidirectional {
            protocol ah;
            spi 256;
            authentication {
                algorithm hmac-md5-96;
                key ascii-text "$9$AP5Hp1RcylMLxSygoZUHk1REhKMVwY2oJx7jHq.zF69A0OR"; ## SECRET-
DATA
            }
            encryption {
                algorithm des;
                key ascii-text "$9$AP5Hp1RcylMLxSygoZUHk1REhKMVwY2oJx7jHq.zF69A0OR"; ## SECRET-
DATA
            }
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Enabling IPsec Authentication for an OSPF Interface

CLI Quick Configuration

To quickly apply a manual SA used for IPsec authentication to an OSPF interface, copy the following command, paste it into a text file, change any details necessary to match your network configuration, copy and paste the command into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
[edit]
set protocols ospf area 0.0.0.0 interface so-0/2/0 ipsec-sa sa1
```

Step-by-Step Procedure

To enable IPsec authentication for an OSPF interface:

1. Create an OSPF area.

   To specify OSPFv3, include the `ospf3` statement at the `[edit protocols]` hierarchy level.

   ```
   [edit]
   user@host# edit protocols ospf area 0.0.0.0
   ```

2. Specify the interface.

```
[edit protocols ospf area 0.0.0.0]
user@host# edit interface so-0/2/0
```

3. Apply the IPsec manual SA.

```
[edit protocols ospf area 0.0.0.0 interface so-0/2/0.0]
user@host# set ipsec-sa sa1
```

Results

Confirm your configuration by entering the **show ospf interface detail** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

To confirm your OSPFv3 configuration, enter the **show protocols ospf3** command.

```
[edit]
user@host# show protocols ospf
area 0.0.0.0 {
    interface so-0/2/0.0 {
         ipsec-sa sa1;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the IPsec Security Association Settings

Purpose

Verify the configured IPsec security association settings. Verify the following information:

- The Security association field displays the name of the configured security association.
- The SPI field displays the value you configured.
- The Mode field displays transport mode.
- The Type field displays manual as the type of security association.

Action

From operational mode, enter the **show ospf interface detail** command.

Verifying the IPsec Security Association on the OSPF Interface

Purpose

Verify that the IPsec security association that you configured has been applied to the OSPF interface. Confirm that the IPsec SA name field displays the name of the configured IPsec security association.

Action

From operational mode, enter the **show ospf interface detail** command for OSPF, and enter the **show ospf3 interface detail** command for OSPFv3.

SEE ALSO

Understanding IPsec SA Configuration for Group VPNv1 | 715

Configuring IPsec VPN Using the VPN Wizard

The VPN Wizard enables you to perform basic IPsec VPN configuration, including both Phase 1 and Phase 2. For more advanced configuration, use the J-Web interface or the CLI. This feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

To configure IPsec VPN using the VPN Wizard:

1. Select `Configure>Device Setup>VPN` in the J-Web interface.
2. Click the Launch VPN Wizard button.
3. Follow the wizard prompts.

The upper left area of the wizard page shows where you are in the configuration process. The lower left area of the page shows field-sensitive help. When you click a link under the Resources heading, the document opens in your browser. If the document opens in a new tab, be sure to close only the tab (not the browser window) when you close the document.

SEE ALSO

IPsec Overview | 20

Internet Key Exchange | 10

Example: Configuring a Hub-and-Spoke VPN

IN THIS SECTION

This example shows how to configure a hub-and-spoke IPsec VPN for an enterprise-class deployment. For site-to-site IPSec VPN with IKEv1 and IKEv2, see Route-Based IPsec VPN with IKEv1 and Route-Based IPsec VPN with IKEv1 respectively.

Requirements

This example uses the following hardware:

- SRX240 device
- SRX5800 device
- SSG140 device

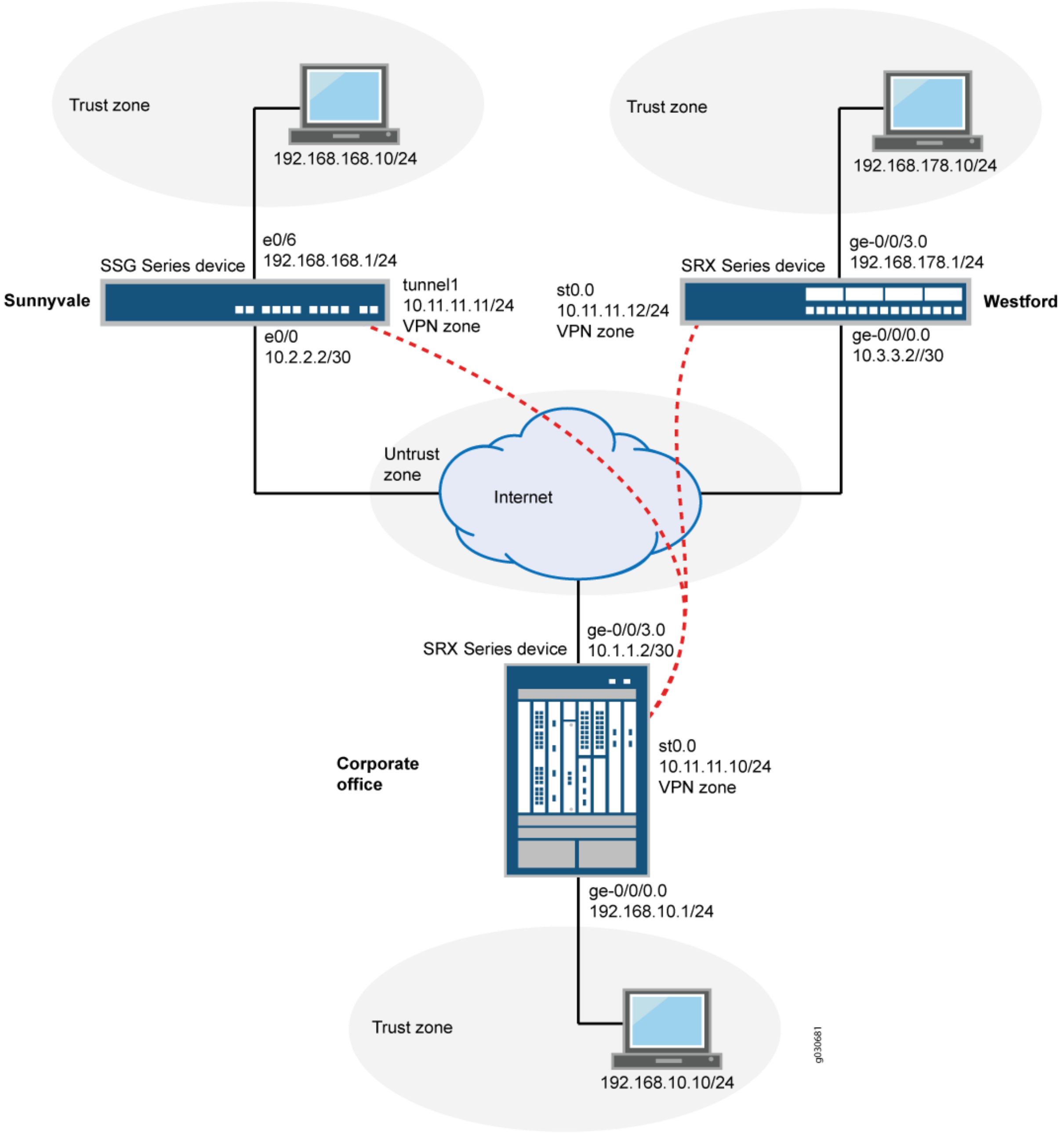
Before you begin, read "IPsec Overview" on page 20.

Overview

This example describes how to configure a hub-and-spoke VPN typically found in branch deployments. The hub is the corporate office, and there are two spokes—a branch office in Sunnyvale, California, and a branch office in Westford, Massachusetts. Users in the branch offices will use the VPN to securely transfer data with the corporate office.

Figure 25 on page 211 shows an example of a hub-and-spoke VPN topology. In this topology, an SRX5800 device is located at the corporate office. An SRX Series Firewall is located at the Westford branch, and an SSG140 device is located at the Sunnyvale branch.

Figure 25: Hub-and-Spoke VPN Topology



In this example, you configure the corporate office hub, the Westford spoke, and the Sunnyvale spoke. First you configure interfaces, IPv4 static and default routes, security zones, and address books. Then you configure IKE Phase 1 and IPsec Phase 2 parameters, and bind the st0.0 interface to the IPsec VPN. On the hub, you configure st0.0 for multipoint and add a static NHTB table entry for the Sunnyvale spoke. Finally, you configure security policy and TCP-MSS parameters. See Table 21 on page 212 through Table 25 on page 220 for specific configuration parameters used in this example.

Table 21: Interface, Security Zone, and Address Book Information

Hub or Spoke	Feature	Name	Configuration Parameters
Hub	Interfaces	ge-0/0/0.0	192.168.10.1/24
		ge-0/0/3.0	10.1.1.2/30
		st0	10.11.11.10/24
Spoke	Interfaces	ge-0/0/0.0	10.3.3.2/30
		ge-0/0/3.0	192.168.178.1/24
		st0	10.11.11.12/24
Hub	Security zones	trust	• All system services are allowed. • The ge-0/0/0.0 interface is bound to this zone.
		untrust	• IKE is the only allowed system service. • The ge-0/0/3.0 interface is bound to this zone.
		vpn	The st0.0 interface is bound to this zone.

Table 21: Interface, Security Zone, and Address Book Information *(Continued)*

Hub or Spoke	Feature	Name	Configuration Parameters
Spoke	Security zones	trust	• All system services are allowed. • The ge-0/0/3.0 interface is bound to this zone.
		untrust	• IKE is the only allowed system service. • The ge-0/0/0.0 interface is bound to this zone.
		vpn	The st0.0 interface is bound to this zone.
Hub	Address book entries	local-net	• This address is for the trust zone's address book. • The address for this address book entry is 192.168.10.0/24.
		sunnyvale-net	• This address book is for the vpn zone's address book. • The address for this address book entry is 192.168.168.0/24.

Table 21: Interface, Security Zone, and Address Book Information *(Continued)*

Hub or Spoke	Feature	Name	Configuration Parameters
		westford-net	• This address is for the vpn zone's address book. • The address for this address book entry is 192.168.178.0/24.
Spoke	Address book entries	local-net	• This address is for the trust zone's address book. • The address for this address book entry is 192.168.168.178.0/24.
		corp-net	• This address is for the vpn zone's address book. • The address for this address book entry is 192.168.10.0/24.
		sunnyvale-net	• This address is for the vpn zone's address book. • The address for this address book entry is 192.168.168.0/24.

Table 22: IKE Phase 1 Configuration Parameters

Hub or Spoke	Feature	Name	Configuration Parameters
Hub	Proposal	ike-phase1-proposal	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: aes-128-cbc
	Policy	ike-phase1-policy	• Mode: main • Proposal reference: ike-phase1-proposal • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
	Gateway	gw-westford	• IKE policy reference: ike-phase1-policy • External interface: ge-0/0/3.0 • Gateway address: 10.3.3.2

Table 22: IKE Phase 1 Configuration Parameters *(Continued)*

Hub or Spoke	Feature	Name	Configuration Parameters
		gw-sunnyvale	• IKE policy reference: ike-phase1-policy • External interface: ge-0/0/3.0 • Gateway address: 10.2.2.2
Spoke	Proposal	ike-phase1-proposal	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: aes-128-cbc
	Policy	ike-phase1-policy	• Mode: main • Proposal reference: ike-phase1-proposal • IKE Phase 1 policy authentication method: pre-shared-key ascii-text

Table 22: IKE Phase 1 Configuration Parameters *(Continued)*

Hub or Spoke	Feature	Name	Configuration Parameters
	Gateway	gw-corporate	• IKE policy reference: ike-phase1-policy • External interface: ge-0/0/0.0 • Gateway address: 10.1.1.2

Table 23: IPsec Phase 2 Configuration Parameters

Hub or Spoke	Feature	Name	Configuration Parameters
Hub	Proposal	ipsec-phase2-proposal	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: aes-128-cbc
	Policy	ipsec-phase2-policy	• Proposal reference: ipsec-phase2-proposal • PFS: Diffie-Hellman group2
	VPN	vpn-sunnyvale	• IKE gateway reference: gw-sunnyvale • IPsec policy reference: ipsec-phase2-policy • Bind to interface: st0.0
		vpn-westford	• IKE gateway reference: gw-westford • IPsec policy reference: ipsec-phase2-policy • Bind to interface: st0.0

Table 23: IPsec Phase 2 Configuration Parameters *(Continued)*

Hub or Spoke	Feature	Name	Configuration Parameters
Spoke	Proposal	ipsec-phase2-proposal	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: aes-128-cbc
	Policy	ipsec-phase2-policy	• Proposal reference: ipsec-phase2-proposal • PFS: Diffie-Hellman group2
	VPN	vpn-corporate	• IKE gateway reference: gw-corporate • IPsec policy reference: ipsec-phase2-policy • Bind to interface: st0.0

Table 24: Security Policy Configuration Parameters

Hub or Spoke	Purpose	Name	Configuration Parameters
Hub	The security policy permits traffic from the trust zone to the vpn zone.	local-to-spokes	• Match criteria: • source-address local-net • destination-address sunnyvale-net • destination-address westford-net • application any

Table 24: Security Policy Configuration Parameters *(Continued)*

Hub or Spoke	Purpose	Name	Configuration Parameters
	The security policy permits traffic from the vpn zone to the trust zone.	spokes-to-local	Match criteria: • source-address sunnyvale-net • source-address westford-net • destination-address local-net • application any
	The security policy permits intrazone traffic.	spoke-to-spoke	Match criteria: • source-address any • destination-address any • application any
Spoke	The security policy permits traffic from the trust zone to the vpn zone.	to-corp	• Match criteria: • source-address local-net • destination-address corp-net • destination-address sunnyvale-net • application any
	The security policy permits traffic from the vpn zone to the trust zone.	from-corp	Match criteria: • source-address corp-net • source-address sunnyvale-net • destination-address local-net • application any

Table 24: Security Policy Configuration Parameters *(Continued)*

Hub or Spoke	Purpose	Name	Configuration Parameters
	The security policy permits traffic from the untrust zone to the trust zone.	permit-any	Match criteria: • source-address any • source-destination any • application any • Permit action: source-nat interface By specifying `source-nat interface`, the SRX Series Firewall translates the source IP address and port for outgoing traffic, using the IP address of the egress interface as the source IP address and a random high-number port for the source port.

Table 25: TCP-MSS Configuration Parameters

Purpose	Configuration Parameters
TCC-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the MTU limits on a network. For VPN traffic, the IPsec encapsulation overhead, along with the IP and frame overhead, can cause the resulting ESP packet to exceed the MTU of the physical interface, which causes fragmentation. Fragmentation results in increased use of bandwidth and device resources. The value of 1350 is a recommended starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.	MSS value: 1350

Configuration

IN THIS SECTION

Configuring Basic Network, Security Zone, and Address Book Information for the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 192.168.10.1/24
 set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
 set interfaces st0 unit 0 family inet address 10.11.11.10/24
 set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
 set routing-options static route 192.168.168.0/24 next-hop 10.11.11.11
 set routing-options static route 192.168.178.0/24 next-hop 10.11.11.12
 set security zones security-zone untrust interfaces ge-0/0/3.0
 set security zones security-zone untrust host-inbound-traffic system-services ike
 set security zones security-zone trust interfaces ge-0/0/0.0
 set security zones security-zone trust host-inbound-traffic system-services all
 set security zones security-zone vpn interfaces st0.0
 set security address-book book1 address local-net 192.168.10.0/24
 set security address-book book1 attach zone trust
 set security address-book book2 address sunnyvale-net 192.168.168.0/24
set security address-book book2 address westford-net 192.168.178.0/24
set security address-book book2 attach zone vpn
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure basic network, security zone, and address book information for the hub:

1. Configure Ethernet interface information.

```
[edit]
user@hub# set interfaces ge-0/0/0 unit 0 family inet address 192.168.10.1/24
user@hub# set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
user@hub# set interfaces st0 unit 0 family inet address 10.11.11.10/24
```

2. Configure static route information.

```
[edit]
user@hub# set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
user@hub# set routing-options static route 192.168.168.0/24 next-hop 10.11.11.11
user@hub# set routing-options static route 192.168.178.0/24 next-hop 10.11.11.12
```

3. Configure the untrust security zone.

```
[edit ]
user@hub# set security zones security-zone untrust
```

4. Assign an interface to the untrust security zone.

```
[edit security zones security-zone untrust]
user@hub# set interfaces ge-0/0/3.0
```

5. Specify allowed system services for the untrust security zone.

```
[edit security zones security-zone untrust]
user@hub# set host-inbound-traffic system-services ike
```

6. Configure the trust security zone.

```
[edit]
user@hub# edit security zones security-zone trust
```

7. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@hub# set interfaces ge-0/0/0.0
```

8. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@hub# set host-inbound-traffic system-services all
```

9. Create an address book and attach a zone to it.

```
[edit security address-book book1]
user@hub# set address local-net 10.10.10.0/24
user@hub# set attach zone trust
```

10. Configure the vpn security zone.

```
[edit]
user@hub# edit security zones security-zone vpn
```

11. Assign an interface to the vpn security zone.

```
[edit security zones security-zone vpn]
user@hub# set interfaces st0.0
```

12. Create another address book and attach a zone to it.

```
[edit security address-book book2]
user@hub# set address sunnyvale-net 192.168.168.0/24
user@hub# set address westford-net 192.168.178.0/24
user@hub# set attach zone vpn
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security zones`, and `show security address-book` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 192.168.10.1/24;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 10.1.1.2/30
        }
    }
}
st0{
    unit 0 {
        family inet {
            address 10.11.11.10/24
        }
    }
}
```

```
[edit]
user@hub# show routing-options
```

```
static {
    route 0.0.0.0/0 next-hop 10.1.1.1;
    route 192.168.168.0/24 next-hop 10.11.11.11;
    route 192.168.178.0/24 next-hop 10.11.11.12;
}
```

```
[edit]
user@hub# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone vpn {
    host-inbound-traffic {
    }
    interfaces {
        st0.0;
    }
}
[edit]
user@hub# show security address-book
book1 {
    address local-net 10.10.10.0/24;
    attach {
        zone trust;
    }
}
    book2 {
        address sunnyvale-net 192.168.168.0/24;
        address westford-net 192.168.178.0/24;
        attach {
            zone vpn;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE for the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposal dh-group group2
set security ike proposal ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-phase1-policy mode main
set security ike policy ike-phase1-policy proposals ike-phase1-proposal
set security ike policy ike-phase1-policy pre-shared-key ascii-text “$ABC123”
set security ike gateway gw-westford external-interface ge-0/0/3.0
set security ike gateway gw-westford ike-policy ike-phase1-policy
set security ike gateway gw-westford address 10.3.3.2
set security ike gateway gw-sunnyvale external-interface ge-0/0/3.0
set security ike gateway gw-sunnyvale ike-policy ike-phase1-policy
set security ike gateway gw-sunnyvale address 10.2.2.2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE for the hub:

1. Create the IKE Phase 1 proposal.

```
[edit security ike]
user@hub# set proposal ike-phase1-proposal
```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set authentication-method pre-shared-keys
```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set authentication-algorithm sha1
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set encryption-algorithm aes-128-cbc
```

6. Create an IKE Phase 1 policy.

```
[edit security ike]
user@hub# set policy ike-phase1-policy
```

7. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike-phase1-policy]
user@hub# set mode main
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ike-phase1-policy]
user@hub# set proposals ike-phase1-proposal
```

9. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike-phase1-policy]
user@hub# set pre-shared-key ascii-text “$ABC123”
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@hub# set gateway gw-westford external-interface ge-0/0/3.0
```

11. Define the IKE Phase 1 policy reference.

```
[edit security ike]
user@hub# set gateway gw-westford ike-policy ike-phase1-policy
```

12. Define the IKE Phase 1 gateway address.

```
[edit security ike]
user@hub# set  gateway gw-westford address 10.3.3.2
```

13. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@hub# set gateway gw-sunnyvale external-interface ge-0/0/3.0
```

14. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway]
user@hub# set gateway gw-sunnyvale ike-policy ike-phase1-policy
```

15. Define the IKE Phase 1 gateway address.

    ```
    [edit security ike gateway]
    user@hub# set gateway gw-sunnyvale address 10.2.2.2
    ```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# show security ike
proposal ike-phase1-proposal {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
policy ike-phase1-policy {
    mode main;
    proposals ike-phase1-proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway gw-sunnyvale {
    ike-policy ike-phase1-policy;
    address 10.2.2.2;
    external-interface ge-0/0/3.0;
}
gateway gw-westford {
    ike-policy ike-phase1-policy;
    address 10.3.3.2;
    external-interface ge-0/0/3.0;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec for the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipsec-phase2-proposal protocol esp
set security ipsec proposal ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipsec-phase2-policy proposals ipsec-phase2-proposal
set security ipsec policy ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn vpn-westford ike gateway gw-westford
set security ipsec vpn vpn-westford ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn vpn-westford bind-interface st0.0
set security ipsec vpn vpn-sunnyvale ike gateway gw-sunnyvale
set security ipsec vpn vpn-sunnyvale ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn vpn-sunnyvale bind-interface st0.0
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.11 ipsec-vpn vpn-sunnyvale
set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.12 ipsec-vpn vpn-westford
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec for the hub:

1. Create an IPsec Phase 2 proposal.

   ```
   [edit]
   user@hub# set security ipsec proposal ipsec-phase2-proposal
   ```

2. Specify the IPsec Phase 2 proposal protocol.

   ```
   [edit security ipsec proposal ipsec-phase2-proposal]
   user@hub# set protocol esp
   ```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@hub# set authentication-algorithm hmac-sha1-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@hub# set encryption-algorithm aes-128-cbc
```

5. Create the IPsec Phase 2 policy.

```
[edit security ipsec]
user@hub# set policy ipsec-phase2-policy
```

6. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec-phase2-policy]
user@hub# set proposals ipsec-phase2-proposal
```

7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.

```
[edit security ipsec policy ipsec-phase2-policy]
user@host# set perfect-forward-secrecy keys group2
```

8. Specify the IKE gateways.

```
[edit security ipsec]
user@hub# set vpn vpn-westford ike gateway gw-westford
user@hub# set vpn vpn-sunnyvale ike gateway gw-sunnyvale
```

9. Specify the IPsec Phase 2 policies.

   ```
   [edit security ipsec]
   user@hub# set vpn vpn-westford ike ipsec-policy ipsec-phase2-policy
   user@hub# set vpn vpn-sunnyvale ike ipsec-policy ipsec-phase2-policy
   ```

10. Specify the interface to bind.

    ```
    [edit security ipsec]
    user@hub# set vpn vpn-westford bind-interface st0.0
    user@hub# set vpn vpn-sunnyvale bind-interface st0.0
    ```

11. Configure the st0 interface as multipoint.

    ```
    [edit]
    user@hub# set interfaces st0 unit 0 multipoint
    ```

12. Add static NHTB table entries for the Sunnyvale and Westford offices.

    ```
    [edit]
    user@hub# set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.11 ipsec-vpn vpn-
    sunnyvale
    user@hub# set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.12 ipsec-vpn vpn-
    westford
    ```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# show security ipsec
proposal ipsec-phase2-proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
}
```

```
policy ipsec-phase2-policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec-phase2-proposal;
}
vpn vpn-sunnyvale {
    bind-interface st0.0;
    ike {
        gateway gw-sunnyvale;
        ipsec-policy ipsec-phase2-policy;
    }
}
vpn vpn-westford {
    bind-interface st0.0;
    ike {
        gateway gw-westford;
        ipsec-policy ipsec-phase2-policy;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Security Policies for the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security policies from-zone trust to-zone vpn policy local-to-spokes match source-address
local-net
set security policies from-zone trust to-zone vpn policy local-to-spokes match destination-
address sunnyvale-net
set security policies from-zone trust to-zone vpn policy local-to-spokes match destination-
address westford-net
set security policies from-zone trust to-zone vpn policy local-to-spokes match application any
set security policies from-zone trust to-zone vpn policy local-to-spokes then permit
set security policies from-zone vpn to-zone trust policy spokes-to-local match source-address
sunnyvale-net
set security policies from-zone vpn to-zone trust policy spokes-to-local match source-address
```

```
westford-net
set security policies from-zone vpn to-zone trust policy spokes-to-local match destination-
address local-net
set security policies from-zone vpn to-zone trust policy spokes-to-local match application any
set security policies from-zone vpn to-zone trust policy spokes-to-local then permit
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke match source-address any
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke match destination-address
any
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke match application any
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke then permit
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure security policies for the hub:

1. Create the security policy to permit traffic from the trust zone to the vpn zone.

```
[edit security policies from-zone trust to-zone vpn]
user@hub# set policy local-to-spokes match source-address local-net
user@hub# set policy local-to-spokes match destination-address sunnyvale-net
user@hub# set policy local-to-spokes match destination-address westford-net
user@hub# set policy local-to-spokes match application any
user@hub# set policy local-to-spokes then permit
```

2. Create the security policy to permit traffic from the vpn zone to the trust zone.

```
[edit security policies from-zone vpn to-zone trust]
user@hub# set policy spokes-to-local match source-address sunnyvale-net
user@hub# set policy spokes-to-local match source-address westford-net
user@hub# set policy spokes-to-local match destination-address local-net
user@hub# set policy spokes-to-local match application any
user@hub# set policy spokes-to-local then permit
```

3. Create the security policy to permit intrazone traffic.

```
[edit security policies from-zone vpn to-zone vpn]
user@hub# set policy spoke-to-spoke match source-address any
```

```
user@hub# set policy spoke-to-spoke match destination-address any
user@hub# set policy spoke-to-spoke match application any
user@hub# set policy spoke-to-spoke then permit
```

Results

From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# show security policies
from-zone trust to-zone vpn {
    policy local-to-spokes {
        match {
            source-address local-net;
            destination-address [ sunnyvale-net westford-net ];
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy spokes-to-local {
        match {
            source-address [ sunnyvale-net westford-net ];
            destination-address local-net;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn to-zone vpn {
    policy spoke-to-spoke {
        match {
            source-address any;
            destination-address any;
```

```
                application any;
            }
            then {
                permit;
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring TCP-MSS for the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security flow tcp-mss ipsec-vpn mss 1350
```

Step-by-Step Procedure

To configure TCP-MSS information for the hub:

1. Configure TCP-MSS information.

   ```
   [edit]
   user@hub# set security flow tcp-mss ipsec-vpn mss 1350
   ```

Results

From configuration mode, confirm your configuration by entering the `show security flow` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# show security flow
tcp-mss {
    ipsec-vpn {
```

```
            mss 1350;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Basic Network, Security Zone, and Address Book Information for the Westford Spoke

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.3.3.2/30
set interfaces ge-0/0/3 unit 0 family inet address 192.168.178.1/24
set interfaces st0 unit 0 family inet address 10.11.11.12/24
set routing-options static route 0.0.0.0/0 next-hop 10.3.3.1
set routing-options static route 10.10.10.0/24 next-hop 10.11.11.10
set routing-options static route 192.168.168.0/24 next-hop 10.11.11.10
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone trust interfaces ge-0/0/3.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone vpn interfaces st0.0
set security address-book book1 address local-net 192.168.178.0/24
set security address-book book1 attach zone trust
set security address-book book2 address corp-net 10.10.10.0/24
set security address-book book2 address sunnyvale-net 192.168.168.0/24
set security address-book book2 attach zone vpn
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure basic network, security zone, and address book information for the Westford spoke:

1. Configure Ethernet interface information.

```
[edit]
user@spoke# set interfaces ge-0/0/0 unit 0 family inet address 10.3.3.2/30
user@spoke# set interfaces ge-0/0/3 unit 0 family inet address 192.168.178.1/24
user@spoke# set interfaces st0 unit 0 family inet address 10.11.11.12/24
```

2. Configure static route information.

```
[edit]
user@spoke# set routing-options static route 0.0.0.0/0 next-hop 10.3.3.1
user@spoke# set routing-options static route 10.10.10.0/24 next-hop 10.11.11.10
user@spoke# set routing-options static route 192.168.168.0/24 next-hop 10.11.11.10
```

3. Configure the untrust security zone.

```
[edit]
user@spoke# set security zones security-zone untrust
```

4. Assign an interface to the security zone.

```
[edit security zones security-zone untrust]
user@spoke# set interfaces ge-0/0/0.0
```

5. Specify allowed system services for the untrust security zone.

```
[edit security zones security-zone untrust]
user@spoke# set host-inbound-traffic system-services ike
```

6. Configure the trust security zone.

```
[edit]
user@spoke# edit security zones security-zone trust
```

7. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@spoke# set interfaces ge-0/0/3.0
```

8. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@spoke# set host-inbound-traffic system-services all
```

9. Configure the vpn security zone.

```
[edit]
user@spoke# edit security zones security-zone vpn
```

10. Assign an interface to the vpn security zone.

```
[edit security zones security-zone vpn]
user@spoke# set interfaces st0.0
```

11. Create an address book and attach a zone to it.

```
[edit security address-book book1]
user@spoke# set address local-net 192.168.178.0/24
user@spoke# set attach zone trust
```

12. Create another address book and attach a zone to it.

```
[edit security address-book book2]
user@spoke# set address corp-net 10.10.10.0/24
user@spoke# set address sunnyvale-net 192.168.168.0/24
user@spoke# set attach zone vpn
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, **show security zones**, and **show security address-book** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@spoke# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.3.3.2/30;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.168.178.1/24;
        }
    }
}
st0 {
    unit 0 {
        family inet {
            address 10.11.11.10/24;
        }
    }
}
```

```
[edit]
user@spoke# show routing-options
static {
    route 0.0.0.0/0 next-hop 10.3.3.1;
    route 192.168.168.0/24 next-hop 10.11.11.10;
    route 10.10.10.0/24 next-hop 10.11.11.10;
}
```

```
[edit]
user@spoke# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone vpn {
    interfaces {
        st0.0;
    }
}
[edit]
user@spoke# show security address-book
book1 {
    address corp-net 10.10.10.0/24;
    attach {
        zone trust;
    }
}
    book2 {
        address local-net 192.168.178.0/24;
        address sunnyvale-net 192.168.168.0/24;
        attach {
            zone vpn;
        }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring IKE for the Westford Spoke

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposal dh-group group2
set security ike proposal ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-phase1-policy mode main
set security ike policy ike-phase1-policy proposals ike-phase1-proposal
set security ike policy ike-phase1-policy pre-shared-key ascii-text “$ABC123”
set security ike gateway gw-corporate external-interface ge-0/0/0.0
set security ike gateway gw-corporate ike-policy ike-phase1-policy
set security ike gateway gw-corporate address 10.1.1.2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE for the Westford spoke:

1. Create the IKE Phase 1 proposal.

   ```
   [edit security ike]
   user@spoke# set proposal ike-phase1-proposal
   ```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal ike-phase1-proposal]
user@spoke# set authentication-method pre-shared-keys
```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike-phase1-proposal]
user@spoke# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@spoke# set authentication-algorithm sha1
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@spoke# set encryption-algorithm aes-128-cbc
```

6. Create an IKE Phase 1 policy.

```
[edit security ike]
user@spoke# set policy ike-phase1-policy
```

7. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike-phase1-policy]
user@spoke# set mode main
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ike-phase1-policy]
user@spoke# set proposals ike-phase1-proposal
```

9. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike-phase1-policy]
user@spoke# set pre-shared-key ascii-text “$ABC123”
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@spoke# set gateway gw-corporate external-interface ge-0/0/0.0
```

11. Define the IKE Phase 1 policy reference.

```
[edit security ike]
user@spoke# set gateway gw-corporate ike-policy ike-phase1-policy
```

12. Define the IKE Phase 1 gateway address.

```
[edit security ike]
user@spoke# set gateway gw-corporate address 10.1.1.2
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@spoke# show security ike
proposal ike-phase1-proposal {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
policy ike-phase1-policy {
    mode main;
    proposals ike-phase1-proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway gw-corporate {
    ike-policy ike-phase1-policy;
    address 10.1.1.2;
    external-interface ge-0/0/0.0;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec for the Westford Spoke

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipsec-phase2-proposal protocol esp
set security ipsec proposal ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipsec-phase2-policy proposals ipsec-phase2-proposal
set security ipsec policy ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn vpn-corporate ike gateway gw-corporate
set security ipsec vpn vpn-corporate ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn vpn-corporate bind-interface st0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec for the Westford spoke:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@spoke# set security ipsec proposal ipsec-phase2-proposal
```

2. Specify the IPsec Phase 2 proposal protocol.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@spoke# set protocol esp
```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@spoke# set authentication-algorithm hmac-sha1-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@spoke# set encryption-algorithm aes-128-cbc
```

5. Create the IPsec Phase 2 policy.

```
[edit security ipsec]
user@spoke# set policy ipsec-phase2-policy
```

6. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec-phase2-policy]
user@spoke# set proposals ipsec-phase2-proposal
```

7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.

```
[edit security ipsec policy ipsec-phase2-policy]
user@host# set perfect-forward-secrecy keys group2
```

8. Specify the IKE gateway.

```
[edit security ipsec]
user@spoke# set vpn vpn-corporate ike gateway gw-corporate
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@spoke# set vpn vpn-corporate ike ipsec-policy ipsec-phase2-policy
```

10. Specify the interface to bind.

```
[edit security ipsec]
user@spoke# set vpn vpn-corporate bind-interface st0.0
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@spoke# show security ipsec
proposal ipsec-phase2-proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
}
policy ipsec-phase2-policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec-phase2-proposal;
}
vpn vpn-corporate {
    bind-interface st0.0;
    ike {
        gateway gw-corporate;
        ipsec-policy ipsec-phase2-policy;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Security Policies for the Westford Spoke

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security policies from-zone trust to-zone vpn policy to-corporate match source-address local-
net
set security policies from-zone trust to-zone vpn policy to-corporate match destination-address
corp-net
set security policies from-zone trust to-zone vpn policy to-corporate match destination-address
sunnyvale-net
set security policies from-zone trust to-zone vpn policy to-corporate application any
set security policies from-zone trust to-zone vpn policy to-corporate then permit
set security policies from-zone vpn to-zone trust policy from-corporate match source-address
corp-net
set security policies from-zone vpn to-zone trust policy from-corporate match source-address
sunnyvale-net
set security policies from-zone vpn to-zone trust policy from-corporate match destination-
address local-net
set security policies from-zone vpn to-zone trust policy from-corporate application any
set security policies from-zone vpn to-zone trust policy from-corporate then permit
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure security policies for the Westford spoke:

1. Create the security policy to permit traffic from the trust zone to the vpn zone.

```
[edit security policies from-zone trust to-zone vpn]
user@spoke# set policy to-corp match source-address local-net
user@spoke# set policy to-corp match destination-address corp-net
user@spoke# set policy to-corp match destination-address sunnyvale-net
user@spoke# set policy to-corp match application any
user@spoke# set policy to-corp then permit
```

2. Create the security policy to permit traffic from the vpn zone to the trust zone.

```
[edit security policies from-zone vpn to-zone trust]
user@spoke# set policy spokes-to-local match source-address corp-net
user@spoke# set policy spokes-to-local match source-address sunnyvale-net
user@spoke# set policy spokes-to-local match destination-address local-net
user@spoke# set policy spokes-to-local match application any
user@spoke# set policy spokes-to-local then permit
```

Results

From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@spoke# show security policies
from-zone trust to-zone vpn {
    policy to-corp {
        match {
            source-address local-net;
            destination-address [ sunnyvale-net westford-net ];
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy spokes-to-local {
        match {
            source-address [ sunnyvale-net westford-net ];
            destination-address local-net;
            application any;
        }
        then {
            permit;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring TCP-MSS for the Westford Spoke

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security flow tcp-mss ipsec-vpn mss 1350
```

Step-by-Step Procedure

To configure TCP-MSS for the Westford spoke:

1. Configure TCP-MSS information.

   ```
   [edit]
   user@spoke# set security flow tcp-mss ipsec-vpn mss 1350
   ```

Results

From configuration mode, confirm your configuration by entering the `show security flow` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@spoke# show security flow
tcp-mss {
    ipsec-vpn {
        mss 1350;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Sunnyvale Spoke

CLI Quick Configuration

This example uses an SSG Series device for the Sunnyvale spoke. For reference, the configuration for the SSG Series device is provided. For information about configuring SSG Series devices, see the *Concepts and Examples ScreenOS Reference Guide*, which is located at https://www.juniper.net/documentation.

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set zone name "VPN"
set interface ethernet0/6 zone "Trust"
set interface "tunnel.1" zone "VPN"
set interface ethernet0/6 ip 192.168.168.1/24
set interface ethernet0/6 route
set interface ethernet0/0 ip 10.2.2.2/30
set interface ethernet0/0 route
set interface tunnel.1 ip 10.11.11.11/24
set flow tcp-mss 1350
set address "Trust" "sunnyvale-net" 192.168.168.0 255.255.255.0
set address "VPN" "corp-net" 10.10.10.0 255.255.255.0
set address "VPN" "westford-net" 192.168.178.0 255.255.255.0
set ike gateway "corp-ike" address 10.1.1.2 Main outgoing-interface ethernet0/0 preshare
"395psksecr3t" sec-level standard
set vpn corp-vpn monitor optimized rekey
set vpn "corp-vpn" bind interface tunnel.1
set vpn "corp-vpn" gateway "corp-ike" replay tunnel idletime 0 sec-level standard
set policy id 1 from "Trust" to "Untrust" "ANY" "ANY" "ANY" nat src permit
set policy id 2 from "Trust" to "VPN" "sunnyvale-net" "corp-net" "ANY" permit
set policy id 2
exit
set dst-address "westford-net"
exit
set policy id 3 from "VPN" to "Trust" "corp-net" "sunnyvale-net" "ANY" permit
set policy id 3
set src-address "westford-net"
exit
set route 10.10.10.0/24 interface tunnel.1
set route 192.168.178.0/24 interface tunnel.1
set route 0.0.0.0/0 interface ethernet0/0 gateway 10.2.2.1
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

Before starting the verification process, you need to send traffic from a host in the 192.168.10/24 network to a host in the 192.168.168/24 and 192.168.178/24 networks to bring the tunnels up. For route-based VPNs, you can send traffic initiated from the SRX Series Firewall through the tunnel. We recommend that when testing IPsec tunnels, you send test traffic from a separate device on one side of the VPN to a second device on the other side of the VPN. For example, initiate a ping from 192.168.10.10 to 192.168.168.10.

From operational mode, enter the `show security ike security-associations` command. After obtaining an index number from the command, use the `show security ike security-associations index` *`index_number`* `detail` command.

```
user@hub> show security ike security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  Mode
```

```
6       10.3.3.2        UP     94906ae2263bbd8e  1c35e4c3fc54d6d3  Main
7       10.2.2.2        UP     7e7a1c0367dfe73c  f284221c656a5fbc  Main
```

```
user@hub> show security ike security-associations index 6 detail
IKE peer 10.3.3.2, Index 6,
  Role: Responder, State: UP
  Initiator cookie: 94906ae2263bbd8e,, Responder cookie: 1c35e4c3fc54d6d3
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.1.2:500, Remote: 10.3.3.2:500
  Lifetime: Expires in 3571 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : aes-cbc (128 bits)
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input bytes    :                1128
   Output bytes   :                 988
   Input packets  :                   6
   Output packets :                   5
  Flags: Caller notification sent
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 1
    Negotiation type: Quick mode, Role: Responder, Message ID: 1350777248
    Local: 10.1.1.2:500, Remote: 10.3.3.2:500
    Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    Flags: Caller notification sent, Waiting for done
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.
- Remote Address—Verify that the remote IP address is correct.
- State

  - UP—The Phase 1 SA has been established.

  - DOWN—There was a problem establishing the Phase 1 SA.

- Mode—Verify that the correct mode is being used.

Verify that the following information is correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)

- IKE policy parameters

- Preshared key information

- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations index 1 detail` command lists additional information about the security association with an index number of 1:

- Authentication and encryption algorithms used

- Phase 1 lifetime

- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)

- Initiator and responder role information

  Troubleshooting is best performed on the peer using the responder role.

- Number of IPsec SAs created

- Number of Phase 2 negotiations in progress

Verifying the IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *`index_number`* `detail` command.

```
user@hub> show security ipsec security-associations
  total configured sa: 4
  ID    Gateway          Port  Algorithm         SPI      Life:sec/kb  Mon vsys
  <16384 10.2.2.2         500   ESP:aes-128/sha1   b2fc36f8 3364/ unlim   -   0
  >16384 10.2.2.2         500   ESP:aes-128/sha1   5d73929e 3364/ unlim   -   0
  ID    Gateway          Port  Algorithm         SPI      Life:sec/kb  Mon vsys
  <16385 10.3.3.2         500   ESP:3des/sha1      70f789c6 28756/unlim   -   0
  >16385 10.3.3.2         500   ESP:3des/sha1      80f4126d 28756/unlim   -   0
```

```
user@hub> show security ipsec security-associations index 16385 detail
  Virtual-system: Root
  Local Gateway: 10.1.1.2, Remote Gateway: 10.3.3.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/24)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    DF-bit: clear
    Direction: inbound, SPI: 1895270854, AUX-SPI: 0
    Hard lifetime: Expires in 28729 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 28136 seconds
    Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
    Anti-replay service: enabled, Replay window size: 32

    Direction: outbound, SPI: 2163479149, AUX-SPI: 0
    Hard lifetime: Expires in 28729 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 28136 seconds
    Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
    Anti-replay service: enabled, Replay window size: 32
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 16385. Use this value with the `show security ipsec security-associations index` command to get more information about this particular SA.
- There is one IPsec SA pair using port 500, which indicates that no NAT-traversal is implemented. (NAT-traversal uses port 4500 or another random high-number port.)
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 28756/ unlim value indicates that the Phase 2 lifetime expires in 28756 seconds, and that no lifesize

has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.

- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 16385 detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.
- Another common reason for Phase 2 failure is not specifying the ST interface binding. If IPsec cannot complete, check the kmd log or set trace options.

Verifying Next-Hop Tunnel Bindings

Purpose

After Phase 2 is complete for all peers, verify the next-hop tunnel bindings.

Action

From operational mode, enter the `show security ipsec next-hop-tunnels` command.

```
user@hub> show security ipsec next-hop-tunnels
Next-hop gateway  interface    IPSec VPN name              Flag
10.11.11.11       st0.0        sunnyvale-vpn               Static
10.11.11.12       st0.0        westford-vpn                Auto
```

Meaning

The next-hop gateways are the IP addresses for the st0 interfaces of all remote spoke peers. The next hop should be associated with the correct IPsec VPN name. If no NHTB entry exists, there is no way for the hub device to differentiate which IPsec VPN is associated with which next hop.

The Flag field has one of the following values:

- Static— NHTB was manually configured in the st0.0 interface configurations, which is required if the peer is not an SRX Series Firewall.
- Auto— NHTB was not configured, but the entry was automatically populated into the NHTB table during Phase 2 negotiations between two SRX Series Firewalls

There is no NHTB table for any of the spoke sites in this example. From the spoke perspective, the st0 interface is still a point-to-point link with only one IPsec VPN binding.

Verifying Static Routes for Remote Peer Local LANs

Purpose

Verify that the static route references the spoke peer’s st0 IP address.

Action

From operational mode, enter the `show route` command.

```
user@hub> show route 192.168.168.10
inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.168.0/24   *[Static/5] 00:08:33
                    > to 10.11.11.11 via st0.0
```

```
user@hub> show route 192.168.178.10
inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.178.0/24   *[Static/5] 00:04:04
                    > to 10.11.11.12 via st0.0
```

The next hop is the remote peer’s st0 IP address, and both routes point to st0.0 as the outgoing interface.

Reviewing Statistics and Errors for an IPsec Security Association

Purpose

Review ESP and authentication header counters and errors for an IPsec security association.

Action

From operational mode, enter the `show security ipsec statistics index` command.

```
user@hub> show security ipsec statistics index 16385
ESP Statistics:
  Encrypted bytes:              920
  Decrypted bytes:             6208
  Encrypted packets:              5
  Decrypted packets:             87
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

You can also use the `show security ipsec statistics` command to review statistics and errors for all SAs.

To clear all IPsec statistics, use the `clear security ipsec statistics` command.

Meaning

If you see packet loss issues across a VPN, you can run the `show security ipsec statistics` or `show security ipsec statistics detail` command several times to confirm that the encrypted and decrypted packet counters are incrementing. You should also check whether the other error counters are incrementing.

Testing Traffic Flow Across the VPN

Purpose

Verify the traffic flow across the VPN.

Action

You can use the `ping` command from the SRX Series Firewall to test traffic flow to a remote host PC. Make sure that you specify the source interface so that the route lookup is correct and the appropriate security zones are referenced during policy lookup.

From operational mode, enter the `ping` command.

```
user@hub> ping 192.168.168.10 interface ge-0/0/0 count 5
PING 192.168.168.10 (192.168.168.10): 56 data bytes
64 bytes from 192.168.168.10: icmp_seq=0 ttl=127 time=8.287 ms
64 bytes from 192.168.168.10: icmp_seq=1 ttl=127 time=4.119 ms
64 bytes from 192.168.168.10: icmp_seq=2 ttl=127 time=5.399 ms
64 bytes from 192.168.168.10: icmp_seq=3 ttl=127 time=4.361 ms
64 bytes from 192.168.168.10: icmp_seq=4 ttl=127 time=5.137 ms

--- 192.168.168.10 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.119/5.461/8.287/1.490 ms
```

You can also use the `ping` command from the SSG Series device.

```
user@hub> ping 192.168.10.10 from ethernet0/6
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 192.168.10.10, timeout is 1 seconds from ethernet0/6
!!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=4/4/5 ms
```

```
ssg-> ping 192.168.178.10 from ethernet0/6
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 192.168.178.10, timeout is 1 seconds from
ethernet0/6
!!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=8/8/10 ms
```

Meaning

If the `ping` command fails from the SRX Series or SSG Series device, there might be a problem with the routing, security policies, end host, or encryption and decryption of ESP packets.

SEE ALSO

Understanding Hub-and-Spoke VPNs | 187
Route-Based IPsec VPNs | 394
Example: Configuring a Policy-Based VPN | 268

Release History Table

Release	Description
19.4R1	Starting in Junos OS Release 19.4R1, you can now configure only one dynamic DN attribute among `container-string` and `wildcard-string` at `[edit security ike gateway` *`gateway_name`* `dynamic distinguished-name]` hierarchy. If you try configuring the second attribute after you configure the first attribute, the first attribute is replaced with the second attribute. Before your upgrade your device, you must remove one of the attributes if you have configured both the attributes.
15.1X49-D80	Starting with Junos OS Release 15.1X49-D80, dynamic endpoint VPNs on SRX Series Firewalls support IPv6 traffic on secure tunnels.
12.3X48-D40	Starting with Junos OS Release 12.3X48-D40, Junos OS Release 15.1X49-D70, and Junos OS Release 17.3R1, all dynamic endpoint gateways configured on SRX Series Firewalls that use the same external interface can use different IKE policies, but the IKE policies must use the same IKE proposal.

RELATED DOCUMENTATION

Route-Based VPN with IKEv2 | 421

Comparing Policy-Based and Route-Based VPNs

It is important to understand the differences between policy-based and route-based VPNs and why one might be preferable to the other.

Table 26 on page 261 lists the differences between route-based VPNs and policy-based VPNs.

Table 26: Differences Between Route-Based VPNs and Policy-Based VPNs

Route-Based VPNs	Policy-Based VPNs
With route-based VPNs, a policy does not specifically reference a VPN tunnel.	With policy-based VPN tunnels, a tunnel is treated as an object that, together with source, destination, application, and action, constitutes a tunnel policy that permits VPN traffic.
The policy references a destination address.	In a policy-based VPN configuration, a tunnel policy specifically references a VPN tunnel by name.
The number of route-based VPN tunnels that you create is limited by the number of route entries or the number of st0 interfaces that the device supports, whichever number is lower.	The number of policy-based VPN tunnels that you can create is limited by the number of policies that the device supports.
Route-based VPN tunnel configuration is a good choice when you want to conserve tunnel resources while setting granular restrictions on VPN traffic.	With a policy-based VPN, although you can create numerous tunnel policies referencing the same VPN tunnel, each tunnel policy pair creates an individual IPsec security association (SA) with the remote peer. Each SA counts as an individual VPN tunnel.
With a route-based approach to VPNs, the regulation of traffic is not coupled to the means of its delivery. You can configure dozens of policies to regulate traffic flowing through a single VPN tunnel between two sites, and only one IPsec SA is at work. Also, a route-based VPN configuration allows you to create policies referencing a destination reached through a VPN tunnel in which the action is deny.	In a policy-based VPN configuration, the action must be permit and must include a tunnel.
Route-based VPNs support the exchange of dynamic routing information through VPN tunnels. You can enable an instance of a dynamic routing protocol, such as OSPF, on an st0 interface that is bound to a VPN tunnel.	The exchange of dynamic routing information is not supported in policy-based VPNs.

Table 26: Differences Between Route-Based VPNs and Policy-Based VPNs *(Continued)*

Route-Based VPNs	Policy-Based VPNs
Route-based configurations are used for hub-and-spoke topologies.	Policy-based VPNs cannot be used for hub-and-spoke topologies.
With route-based VPNs, a policy does not specifically reference a VPN tunnel.	When a tunnel does not connect large networks running dynamic routing protocols and you do not need to conserve tunnels or define various policies to filter traffic through the tunnel, a policy-based tunnel is the best choice.
Route-based VPNs do not support remote-access (dial-up) VPN configurations.	Policy-based VPN tunnels are required for remote-access (dial-up) VPN configurations.
Route-based VPNs might not work correctly with some third-party vendors.	Policy-based VPNs might be required if the third party requires separate SAs for each remote subnet.
When the security device does a route lookup to find the interface through which it must send traffic to reach an address, it finds a route via a secure tunnel interface (`st0`) , which is bound to a specific VPN tunnel. With a route-based VPN tunnel, you can consider a tunnel as a means for delivering traffic, and can consider the policy as a method for either permitting or denying the delivery of that traffic.	With a policy-based VPN tunnel, you can consider a tunnel as an element in the construction of a policy.
Route-based VPNs support NAT for st0 interfaces.	Policy-based VPNs cannot be used if NAT is required for tunneled traffic.

Proxy ID is supported for both route-based and policy-based VPNs. Route-based tunnels also offer the usage of multiple traffic selectors also known as multi-proxy ID. A traffic selector is an agreement between IKE peers to permit traffic through a tunnel, if the traffic matches a specified pair of local and remote IP address prefix, source port range, destination port range, and protocol. You define a traffic selector within a specific route-based VPN, which can result in multiple Phase 2 IPsec SAs. Only traffic that conforms to a traffic selector is permitted through an SA. The traffic selector is commonly required when remote gateway devices are non-Juniper Networks devices.

Policy-based VPNs are only supported on SRX5400, SRX5600, and SRX5800 line. Platform support depends on the Junos OS release in your installation.

RELATED DOCUMENTATION

Example: Configuring a Route-Based VPN | 395

Example: Configuring a Policy-Based VPN | 268

Chassis Cluster HA Control Link Encryption

Connect the dedicated control ports on node 0 and node 1. Connect the user defined fabricated ports on node 0 and node 1. To configure two chassis in cluster mode, follow the below steps:

Enable chassis cluster mode on both the nodes, see SRX Series Chassis Cluster Configuration Overview.

1. After enabling the chassis cluster, in the device 1, configure HA link encryption as shown in sample configuration below, commit and reboot. Device 1 needs to be configured with both node0 and node1 HA link encryption configuration before commit and reboot.

```
[edit]
user@host# set groups node0 security ike proposal HA authentication-method pre-shared-keys
user@host# set groups node0 security ike proposal HA dh-group group20
user@host# set groups node0 security ike proposal HA authentication-algorithm sha-256
user@host# set groups node0 security ike proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node0 security ike policy HA proposals HA
user@host# prompt groups node0 security ike policy HA pre-shared-key ascii-text
This Should Be A Strong And Secure Key
Retype This Should Be A Strong And Secure Key
user@host# set groups node0 security ike gateway HA ike-policy HA
user@host# set groups node0 security ike gateway HA version v2-only
user@host# set groups node0 security ipsec proposal HA protocol esp
user@host# set groups node0 security ipsec proposal HA authentication-algorithm hmac-sha1-96
user@host# set groups node0 security ipsec proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node0 security ipsec policy HA perfect-forward-secrecy keys group20
user@host# set groups node0 security ipsec policy HA proposal HA
user@host# set groups node0 security ipsec vpn HA ha-link-encryption
user@host# set groups node0 security ipsec vpn HA ike gateway HA
user@host# set groups node0 security ipsec vpn HA ike ipsec-policy HA
user@host# set groups node1 security ike proposal HA authentication-method pre-shared-keys
```

```
user@host# set groups node1 security ike proposal HA dh-group group20
user@host# set groups node1 security ike proposal HA authentication-algorithm sha-256
user@host# set groups node1 security ike proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node1 security ike policy HA proposals HA
user@host# prompt groups node1 security ike policy HA pre-shared-key ascii-text
New ascii-text(secret): juniper
Retype This Should Be A Strong And Secure Key
user@host# set groups node1 security ike gateway HA ike-policy HA
user@host# set groups node1 security ike gateway HA version v2-only
user@host# set groups node1 security ipsec proposal HA protocol esp
user@host# set groups node1 security ipsec proposal HA authentication-algorithm hmac-sha1-96
user@host# set groups node1 security ipsec proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node1 security ipsec policy HA perfect-forward-secrecy keys group20
user@host# set groups node1 security ipsec policy HA proposals HA
user@host# set groups node1 security ipsec vpn HA ha-link-encryption
user@host# set groups node1 security ipsec vpn HA ike gateway HA
user@host# set groups node1 security ipsec vpn HA ike ipsec-policy HA
user@host# commit
user@host> request system reboot
```

2. To proceed further with device 2 configuration and commit, you need to ensure device 1 and device 2 are not reachable to each other. One way to achieve this is to power off device 1 at this point.
3. After the device 2 is up, configure HA link encryption as shown in sample configuration below on device 2. Device 2 needs to be configured with both node0 and node1 HA link encryption configuration. Commit on node1 (device 2), and finally reboot node1 (device 2).

```
[edit]
user@host# set groups node0 security ike proposal HA authentication-method pre-shared-keys
user@host# set groups node0 security ike proposal HA dh-group group20
user@host# set groups node0 security ike proposal HA authentication-algorithm sha-256
user@host# set groups node0 security ike proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node0 security ike policy HA proposals HA
user@host# prompt groups node0 security ike policy HA pre-shared-key ascii-text
This Should Be A Strong And Secure Key
Retype This Should Be A Strong And Secure Key
user@host# set groups node0 security ike gateway HA ike-policy HA
user@host# set groups node0 security ike gateway HA version v2-only
user@host# set groups node0 security ipsec proposal HA protocol esp
user@host# set groups node0 security ipsec proposal HA authentication-algorithm hmac-sha1-96
user@host# set groups node0 security ipsec proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node0 security ipsec policy HA perfect-forward-secrecy keys group20
user@host# set groups node0 security ipsec policy HA proposal HA
```

```
user@host# set groups node0 security ipsec vpn HA ha-link-encryption
user@host# set groups node0 security ipsec vpn HA ike gateway HA
user@host# set groups node0 security ipsec vpn HA ike ipsec-policy HA
user@host# set groups node1 security ike proposal HA authentication-method pre-shared-keys
user@host# set groups node1 security ike proposal HA dh-group group20
user@host# set groups node1 security ike proposal HA authentication-algorithm sha-256
user@host# set groups node1 security ike proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node1 security ike policy HA proposals HA
user@host# prompt groups node1 security ike policy HA pre-shared-key ascii-text
New ascii-text(secret): juniper
Retype This Should Be A Strong And Secure Key
user@host# set groups node1 security ike gateway HA ike-policy HA
user@host# set groups node1 security ike gateway HA version v2-only
user@host# set groups node1 security ipsec proposal HA protocol esp
user@host# set groups node1 security ipsec proposal HA authentication-algorithm hmac-sha1-96
user@host# set groups node1 security ipsec proposal HA encryption-algorithm aes-256-cbc
user@host# set groups node1 security ipsec policy HA perfect-forward-secrecy keys group20
user@host# set groups node1 security ipsec policy HA proposals HA
user@host# set groups node1 security ipsec vpn HA ha-link-encryption
user@host# set groups node1 security ipsec vpn HA ike gateway HA
user@host# set groups node1 security ipsec vpn HA ike ipsec-policy HA
user@host# commit
user@host> request system reboot
```

NOTE: To enable HA link encryption on node1 in step 3, the other node needs to be in lost state for the commit to go through. So this timing needs to be taken care by you, else step 3 needs to be redone until enabling HA link encryption on node1 commit goes through.

6

CHAPTER

Policy Based VPN

Policy-Based IPsec VPNs

IN THIS SECTION

A policy-based VPN is a configuration in which an IPsec VPN tunnel created between two end points is specified within the policy itself with a policy action for the transit traffic that meets the policy’s match criteria.

Understanding Policy-Based IPsec VPNs

For policy-based IPsec VPNs, a security policy specifies as its action the VPN tunnel to be used for transit traffic that meets the policy’s match criteria. A VPN is configured independent of a policy statement. The policy statement refers to the VPN by name to specify the traffic that is allowed access to the tunnel. For policy-based VPNs, each policy creates an individual IPsec security association (SA) with the remote peer, each of which counts as an individual VPN tunnel. For example, if a policy contains a group source address and a group destination address, whenever one of the users belonging to the address set attempts to communicate with any one of the hosts specified as the destination address, a new tunnel is negotiated and established. Because each tunnel requires its own negotiation process and separate pair of SAs, the use of policy-based IPsec VPNs can be more resource-intensive than route-based VPNs.

Examples of where policy-based VPNs can be used:

- You are implementing a dial-up VPN.
- Policy-based VPNs allow you to direct traffic based on firewall policies.

We recommend that you use route-based VPN when you want to configure a VPN between multiple remote sites. Route-based VPNs can provide the same capabilities as policy-based VPNs.

Limitations:

- Policy-based IPSec VPNs are not supported with IKEv2.

SEE ALSO

- IPsec Overview | 20
- Example: Configuring a Route-Based VPN | 395
- Example: Configuring a Hub-and-Spoke VPN | 209

Example: Configuring a Policy-Based VPN

IN THIS SECTION

This example shows how to configure a policy-based IPsec VPN to allow data to be securely transferred between two sites.

Requirements

This example uses the following hardware:

- Any SRX Series Firewall
  - Updated and revalidated using vSRX Virtual Firewall on Junos OS Release 20.4R1.

NOTE: Are you interested in getting hands-on experience with the topics and operations covered in this guide? Visit the IPsec Policy-Based demonstration in Juniper Networks Virtual Labs and reserve your free sandbox today! You'll find the IPsec VPN Policy-Based sandbox in the Security category.

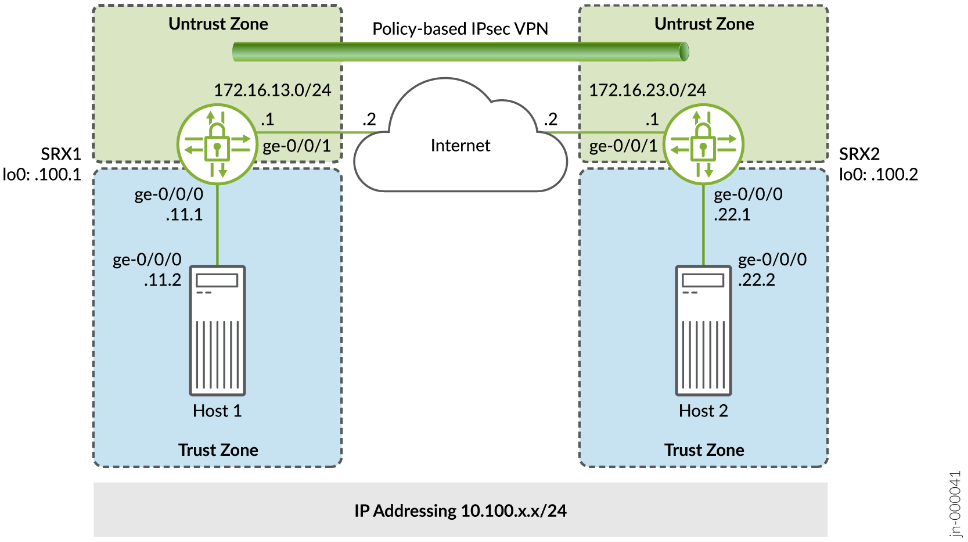
Before you begin, read "IPsec Overview" on page 20.

Overview

In this example, you configure a policy-based VPN on SRX1 and SRX2. Host1 and Host2 use the VPN to send traffic securely over the Internet between both hosts.

Figure 26 on page 269 shows an example of a policy-based VPN topology.

Figure 26: Policy-Based VPN Topology



IKE IPsec tunnel negotiation occurs in two phases. In Phase 1, participants establish a secure channel in which to negotiate the IPsec security association (SA). In Phase 2, participants negotiate the IPsec SA for authenticating traffic that will flow through the tunnel. Just as there are two phases to tunnel negotiation, there are two phases to tunnel configuration.

In this example, you configure interfaces, an IPv4 default route, and security zones. Then you configure IKE Phase 1, IPsec Phase 2, security policy, and TCP-MSS parameters. See Table 27 on page 269 through Table 31 on page 272.

Table 27: Interface, Static Route, and Security Zone Information for SRX1

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	10.100.11.1/24
	ge-0/0/1.0	172.16.13.1/24
Security zones	trust	• The ge-0/0/0.0 interface is bound to this zone.

Table 27: Interface, Static Route, and Security Zone Information for SRX1 *(Continued)*

Feature	Name	Configuration Parameters
	untrust	- The ge-0/0/1.0 interface is bound to this zone.
Static routes	0.0.0.0/0	- The next hop is 172.16.13.2.

Table 28: IKE Phase 1 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	standard	- Authentication method: pre-shared-keys
Policy	IKE-POL	- Mode: main - Proposal reference: standard - IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	IKE-GW	- IKE policy reference: IKE-POL - External interface: ge-0/0/1 - Gateway address: 172.16.23.1

Table 29: IPsec Phase 2 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	standard	- Using default configuration
Policy	IPSEC-POL	- Proposal reference: standard

Table 29: IPsec Phase 2 Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
VPN	VPN-to-Host2	• IKE gateway reference: IKE-GW • IPsec policy reference: IPSEC-POL • establish-tunnels immediately

Table 30: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	VPN-OUT	• Match criteria: • source-address Host1-Net • destination-address Host2-Net • application any • Permit action: tunnel ipsec-vpn VPN-to-Host2
This security policy permits traffic from the untrust zone to the trust zone.	VPN-IN	• Match criteria: • source-address Host2-Net • destination-address Host1-Net • application any • Permit action: tunnel ipsec-vpn VPN-to-Host2

Table 30: Security Policy Configuration Parameters *(Continued)*

Purpose	Name	Configuration Parameters
This security policy permits all traffic from the trust zone to the untrust zone. You must put the VPN-OUT policy before the default-permit security policy. Junos OS performs a security policy lookup starting at the top of the list. If the default-permit policy comes before the VPN-OUT policy, all traffic from the trust zone matches the default-permit policy and is permitted. Thus, no traffic will ever match the VPN-OUT policy.	default-permit	• Match criteria: • source-address any • source-destination any • application any • Action: permit

Table 31: TCP-MSS Configuration Parameters

Purpose	Configuration Parameters
TCP-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the maximum transmission unit (MTU) limits on a network. This is especially important for VPN traffic, as the IPsec encapsulation overhead, along with the IP and frame overhead, can cause the resulting Encapsulating Security Payload (ESP) packet to exceed the MTU of the physical interface, thus causing fragmentation. Fragmentation results in increased use of bandwidth and device resources. We recommend a value of 1350 as the starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.	MSS value: 1350

Configuration

IN THIS SECTION

Configuring Basic Network and Security Zone Information

CLI Quick Configuration

To quickly configure this example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.100.11.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.13.1/24
set interfaces lo0 unit 0 family inet address 10.100.100.1/32
set routing-options static route 0.0.0.0/0 next-hop 172.16.13.2
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do this, see the CLI User Guide.

To configure interface, static route, and security zone information:

1. Configure the interfaces.

```
[edit]
user@SRX1# set interfaces ge-0/0/0 unit 0 family inet address 10.100.11.1/24
user@SRX1# set interfaces ge-0/0/1 unit 0 family inet address 172.16.13.1/24
user@SRX1# set interfaces lo0 unit 0 family inet address 10.100.100.1/32
```

2. Configure the static routes.

   ```
   [edit]
   user@SRX1# set routing-options static route 0.0.0.0/0 next-hop 172.16.13.2
   ```

3. Assign the Internet facing interface to the untrust security zone.

   ```
   [edit security zones security-zone untrust]
   user@SRX1# set interfaces ge-0/0/1.0
   ```

4. Specify the allowed system services for the untrust security zone.

   ```
   [edit security zones security-zone untrust]
   user@SRX1# set host-inbound-traffic system-services ike
   user@SRX1# set host-inbound-traffic system-services ping
   ```

5. Assign the Host1 facing interface to the trust security zone.

   ```
   [edit security zones security-zone trust]
   user@SRX1# set interfaces ge-0/0/0.0
   ```

6. Specify the allowed system services for the trust security zone.

   ```
   [edit security zones security-zone trust]
   user@SRX1# set host-inbound-traffic system-services all
   ```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security zones` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@SRX1# show interfaces
ge-0/0/0 {
    unit 0 {
```

```
        family inet {
            address 10.100.11.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 172.16.13.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.100.100.1/32;
        }
    }
}
```

```
[edit]
user@SRX1# show routing-options
static {
    route 0.0.0.0/0 next-hop 172.16.13.2;
}
```

```
[edit]
user@SRX1# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
```

Configuring IKE

CLI Quick Configuration

To quickly configure this example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal standard authentication-method pre-shared-keys
set security ike policy IKE-POL mode main
set security ike policy IKE-POL proposals standard
set security ike policy IKE-POL pre-shared-key ascii-text $ABC123
set security ike gateway IKE-GW ike-policy IKE-POL
set security ike gateway IKE-GW address 172.16.23.1
set security ike gateway IKE-GW external-interface ge-0/0/1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure IKE:

1. Create the IKE proposal.

   ```
   [edit security ike]
   user@SRX1# set proposal standard
   ```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal standard]
user@SRX1# set authentication-method pre-shared-keys
```

3. Create the IKE policy.

```
[edit security ike]
user@SRX1# set policy IKE-POL
```

4. Set the IKE policy mode.

```
[edit security ike policy IKE-POL]
user@SRX1# set mode main
```

5. Specify a reference to the IKE proposal.

```
[edit security ike policy IKE-POL]
user@SRX1# set proposals standard
```

6. Define the IKE policy authentication method.

```
[edit security ike policy IKE-POL]
user@SRX1# set pre-shared-key ascii-text $ABC123
```

7. Create the IKE gateway and define its external interface.

```
[edit security ike gateway IKE-GW]
user@SRX1# set external-interface ge-0/0/1.0
```

8. Define the IKE gateway address.

```
[edit security ike gateway IKE-GW]
user@SRX1# address 172.16.23.1
```

9. Define the IKE policy reference.

   ```
   [edit security ike gateway IKE-GW]
   user@SRX1# set ike-policy IKE-POL
   ```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
proposal standard {
    authentication-method pre-shared-keys;
}
policy IKE-POL {
    mode main;
    proposals standard;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway IKE-GW {
    ike-policy IKE-POL;
    address 172.16.23.1;
    external-interface ge-0/0/1;
}
```

Configuring IPsec

CLI Quick Configuration

To quickly configure this example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal standard
set security ipsec policy IPSEC-POL proposals standard
```

```
set security ipsec vpn VPN-to-Host2 ike gateway IKE-GW
set security ipsec vpn VPN-to-Host2 ike ipsec-policy IPSEC-POL
set security ipsec vpn VPN-to-Host2 establish-tunnels immediately
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure IPsec:

1. Create the IPsec proposal.

   ```
   [edit]
   user@SRX1# set security ipsec proposal standard
   ```

2. Create the IPsec policy.

   ```
   [edit security ipsec]
   user@SRX1# set policy IPSEC-POL
   ```

3. Specify the IPsec proposal reference.

   ```
   [edit security ipsec policy IPSEC-POL]
   user@SRX1# set proposals standard
   ```

4. Specify the IKE gateway.

   ```
   [edit security ipsec]
   user@SRX1# set vpn VPN-to-Host2 ike gateway IKE-GW
   ```

5. Specify the IPsec policy.

   ```
   [edit security ipsec]
   user@SRX1# set vpn VPN-to-Host2 ike ipsec-policy IPSEC-POL
   ```

6. Configure the tunnel to establish immediately.

```
[edit security ipsec]
user@SRX1# set vpn VPN-to-Host2 establish-tunnels immediately
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@SRX1# show security ipsec
proposal standard;
policy IPSEC-POL {
    proposals standard;
}
vpn VPN-to-Host2 {
    ike {
        gateway IKE-GW;
        ipsec-policy IPSEC-POL;
    }
    establish-tunnels immediately;
}
```

Configuring Security Policies

CLI Quick Configuration

To quickly configure this example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set security address-book Host1 address Host1-Net 10.100.11.0/24
set security address-book Host1 attach zone trust
set security address-book Host2 address Host2-Net 10.100.22.0/24
set security address-book Host2 attach zone untrust
set security policies from-zone trust to-zone untrust policy VPN-OUT match source-address Host1-Net
set security policies from-zone trust to-zone untrust policy VPN-OUT match destination-address Host2-Net
set security policies from-zone trust to-zone untrust policy VPN-OUT match application any
set security policies from-zone trust to-zone untrust policy VPN-OUT then permit tunnel ipsec-vpn VPN-to-Host2
set security policies from-zone trust to-zone untrust policy default-permit match source-address any
set security policies from-zone trust to-zone untrust policy default-permit match destination-address any
set security policies from-zone trust to-zone untrust policy default-permit match application any
set security policies from-zone trust to-zone untrust policy default-permit then permit
set security policies from-zone untrust to-zone trust policy VPN-IN match source-address Host2-Net
set security policies from-zone untrust to-zone trust policy VPN-IN match destination-address Host1-Net
set security policies from-zone untrust to-zone trust policy VPN-IN match application any
set security policies from-zone untrust to-zone trust policy VPN-IN then permit tunnel ipsec-vpn VPN-to-Host2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure security policies:

1. Create address book entries for the networks that will be used in the security policies.

   ```
   [edit]
   user@SRX1# set security address-book Host1 address Host1-Net 10.100.11.0/24
   user@SRX1# set security address-book Host1 attach zone trust
   user@SRX1# set security address-book Host2 address Host2-Net 10.100.22.0/24
   user@SRX1# set security address-book Host2 attach zone untrust
   ```

2. Create the security policy to match on traffic from Host1 in the trust zone to Host2 in the untrust zone.

   ```
   [edit security policies from-zone trust to-zone untrust]
   user@SRX1# set policy VPN-OUT match source-address Host1-Net
   ```

```
user@SRX1# set policy VPN-OUT match destination-address Host2-Net
user@SRX1# set policy VPN-OUT match application any
user@SRX1# set policy VPN-OUT then permit tunnel ipsec-vpn VPN-to-Host2
```

3. Create the security policy to permit all other traffic to the Internet from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@SRX1# set policy default-permit match source-address any
user@SRX1# set policy default-permit match destination-address any
user@SRX1# set policy default-permit match application any
user@SRX1# set policy default-permit then permit
```

4. Create a security policy to permit traffic from Host2 in the untrust zone to Host1 in the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@SRX1# set policy VPN-IN match source-address Host2-Net
user@SRX1# set policy VPN-IN match destination-address Host1-Net
user@SRX1# set policy VPN-IN match application any
user@SRX1# set policy VPN-IN then permit tunnel ipsec-vpn VPN-to-Host2
```

Results

From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@SRX1# show security policies
from-zone trust to-zone untrust {
    policy VPN-OUT {
        match {
            source-address Host1-Net;
            destination-address Host2-Net;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-vpn VPN-to-Host2;
                }
            }
        }
    }
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone untrust to-zone trust {
    policy VPN-IN {
        match {
            source-address Host2-Net;
            destination-address Host1-Net;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-vpn VPN-to-Host2;
                }
            }
        }
    }
}
```

Configuring TCP-MSS

CLI Quick Configuration

To quickly configure this example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter commit from configuration mode.

```
set security flow tcp-mss ipsec-vpn mss 1350
```

Step-by-Step Procedure

To configure TCP-MSS information:

1. Configure the TCP-MSS information.

   ```
   [edit]
   user@SRX1# set security flow tcp-mss ipsec-vpn mss 1350
   ```

Results

From configuration mode, confirm your configuration by entering the show security flow command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@SRX1# show security flow
tcp-mss {
    ipsec-vpn {
        mss 1350;
    }
}
```

If you are done configuring the device, enter commit from configuration mode.

Configuring SRX2

CLI Quick Configuration

For reference, the configuration for SRX2 is provided.

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set security ike proposal standard authentication-method pre-shared-keys
set security ike policy IKE-POL mode main
set security ike policy IKE-POL proposals standard
set security ike policy IKE-POL pre-shared-key ascii-text $ABC123
set security ike gateway IKE-GW ike-policy IKE-POL
set security ike gateway IKE-GW address 172.16.13.1
set security ike gateway IKE-GW external-interface ge-0/0/1
set security ipsec proposal standard
set security ipsec policy IPSEC-POL proposals standard
set security ipsec vpn VPN-to-Host1 ike gateway IKE-GW
set security ipsec vpn VPN-to-Host1 ike ipsec-policy IPSEC-POL
set security ipsec vpn VPN-to-Host1 establish-tunnels immediately
set security address-book Host1 address Host1-Net 10.100.11.0/24
set security address-book Host1 attach zone untrust
set security address-book Host2 address Host2-Net 10.100.22.0/24
set security address-book Host2 attach zone trust
set security flow tcp-mss ipsec-vpn mss 1350
set security policies from-zone trust to-zone untrust policy VPN-OUT match source-address Host2-
Net
set security policies from-zone trust to-zone untrust policy VPN-OUT match destination-address
Host1-Net
set security policies from-zone trust to-zone untrust policy VPN-OUT match application any
set security policies from-zone trust to-zone untrust policy VPN-OUT then permit tunnel ipsec-
vpn VPN-to-Host1
set security policies from-zone trust to-zone untrust policy default-permit match source-address
any
set security policies from-zone trust to-zone untrust policy default-permit match destination-
address any
set security policies from-zone trust to-zone untrust policy default-permit match application any
set security policies from-zone trust to-zone untrust policy default-permit then permit
set security policies from-zone untrust to-zone trust policy VPN-IN match source-address Host1-
Net
set security policies from-zone untrust to-zone trust policy VPN-IN match destination-address
Host2-Net
set security policies from-zone untrust to-zone trust policy VPN-IN match application any
set security policies from-zone untrust to-zone trust policy VPN-IN then permit tunnel ipsec-vpn
VPN-to-Host1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet address 10.100.22.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.23.1/24
set interfaces lo0 unit 0 family inet address 10.100.100.2/32
set routing-options static route 0.0.0.0/0 next-hop 172.16.23.2
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the IKE Status

Purpose

Verify the IKE status.

Action

From operational mode, enter the `show security ike security-associations` command. After obtaining an index number from the command, use the `show security ike security-associations index` *`index_number`* `detail` command.

```
user@SRX1> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
1859361 UP     9788fa59c3ee2e2a  0b17e52f34b83aba  Main           172.16.23.1
```

```
user@SRX1> show security ike security-associations index 1859361 detail
IKE peer 172.16.23.1, Index 1859361, Gateway Name: IKE-GW
  Role: Responder, State: UP
  Initiator cookie: 9788fa59c3ee2e2a, Responder cookie: 0b17e52f34b83aba
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 172.16.13.1:500, Remote: 172.16.23.1:500
  Lifetime: Expires in 17567 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
  Remote Access Client Info: Unknown Client
  Peer ike-id: 172.16.23.1
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :                 1740
   Output bytes  :                 1132
   Input  packets:                   15
   Output packets:                    7
   Input  fragmentated packets:       0
   Output fragmentated packets:       0
  IPSec security associations: 4 created, 4 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 172.16.13.1:500, Remote: 172.16.23.1:500
    Local identity: 172.16.13.1
    Remote identity: 172.16.23.1
    Flags: IKE SA is created
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 security associations (SAs). If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.
- Remote Address—Verify that the remote IP address is correct.
- State
  - UP—The Phase 1 SA has been established.
  - DOWN—There was a problem establishing the Phase 1 SA.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations index 1859361 detail` command lists additional information about the security association with an index number of 1859361:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Initiator and responder role information

  Troubleshooting is best performed on the peer using the responder role.
- Number of IPsec SAs created

- Number of Phase 2 negotiations in progress

Verifying the IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *`index_number`* `detail` command.

```
user@SRX1 show security ipsec security-associations
  Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <2      ESP:3des/sha1   ae5afc5a 921/ unlim   -   root 500   172.16.23.1
  >2      ESP:3des/sha1   6388a743 921/ unlim   -   root 500   172.16.23.1
```

```
user@SRX1> show security ipsec security-associations index 2 detail
ID: 2 Virtual-system: root, VPN Name: VPN-to-Host2
  Local Gateway: 172.16.13.1, Remote Gateway: 172.16.23.1
  Local Identity: ipv4_subnet(any:0,[0..7]=10.100.11.0/24)
  Remote Identity: ipv4_subnet(any:0,[0..7]=10.100.22.0/24)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Disabled                             , Policy-name: VPN-OUT
  Port: 500, Nego#: 30, Fail#: 0, Def-Del#: 0 Flag: 0x600829
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Thu Jul 29 2021 14:29:22 -0700: IPSec SA negotiation successfully completed (29 times)
    Thu Jul 29 2021 12:00:30 -0700: IKE SA negotiation successfully completed (4 times)
    Wed Jul 28 2021 15:20:58
    : IPSec SA delete payload received from peer, corresponding IPSec SAs cleared (1 times)
    Wed Jul 28 2021 15:05:13 -0700: IPSec SA negotiation successfully completed (1 times)
    Wed Jul 28 2021 15:05:13
    : Tunnel is ready. Waiting for trigger event or peer to trigger negotiation (1 times)
    Wed Jul 28 2021 15:05:13 -0700: External interface's address received. Information updated
(1 times)
    Wed Jul 28 2021 15:05:13 -0700: External interface's zone received. Information updated (1
times)
    Wed Jul 28 2021 11:17:38
    : Negotiation failed  with error code NO_PROPOSAL_CHOSEN received from peer (1 times)
    Wed Jul 28 2021 09:27:11 -0700: IKE SA negotiation successfully completed (19 times)
    Thu Jul 22 2021 16:34:17 -0700: Negotiation failed with INVALID_SYNTAX error (3 times)
    Thu Jul 22 2021 10:34:55 -0700: IKE SA negotiation successfully completed (1 times)
    Thu Jul 22 2021 10:34:46 -0700: No response from peer. Negotiation failed (16 times)
  Direction: inbound, SPI: ae5afc5a, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 828 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 234 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 6388a743, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 828 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 234 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 2. Use this value with the `show security ipsec security-associations index` command to get more information about this particular SA.

- There is one IPsec SA pair using port 500, which indicates that no NAT-traversal is implemented. (NAT-traversal uses port 4500 or another random high-number port.)

- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 921/ unlim value indicates that the Phase 2 lifetime expires in 921 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.

- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U (up) or D (down) is listed.

- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 2 detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common reasons for a Phase 2 failure. For policy-based VPNs, the proxy ID is derived from the security policy. The local address and remote address are derived from the address book entries, and the service is derived from the application configured for the policy. If Phase 2 fails because of a proxy ID mismatch, you can use the policy to confirm which address book entries are configured. Verify that the addresses match the information being sent. Check the service to ensure that the ports match the information being sent.

Test Traffic Flow Across the VPN

Purpose

Verify the traffic flow across the VPN.

Action

Use the `ping` command from the Host1 device to test traffic flow to Host2.

```
user@Host1> ping 10.100.22.1 rapid count 100
PING 10.100.22.1 (10.100.22.1): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!
--- 10.100.22.1 ping statistics ---
100 packets transmitted, 100 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.300/3.936/8.562/0.720 ms
```

Meaning

If the `ping` command fails from Host1, there might be a problem with the routing, security policies, end host, or encryption and decryption of ESP packets.

Reviewing Statistics and Errors for an IPsec Security Association

Purpose

Review ESP and authentication header counters and errors for an IPsec security association.

Action

From operational mode, enter the `show security ipsec statistics index` *`index_number`* command, using the index number of the VPN for which you want to see statistics.

```
user@SRX1> show security ipsec statistics index 2
ESP Statistics:
  Encrypted bytes:            13600
  Decrypted bytes:             8400
  Encrypted packets:            100
  Decrypted packets:            100
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

You can also use the `show security ipsec statistics` command to review statistics and errors for all SAs.

To clear all IPsec statistics, use the `clear security ipsec statistics` command.

Meaning

If you see packet loss issues across a VPN, you can run the `show security ipsec statistics` command several times to confirm that the encrypted and decrypted packet counters are incrementing. You should also check if the other error counters are incrementing.

SEE ALSO

IPsec Overview | **20**

Example: Configuring a Route-Based VPN | **395**

IPsec Policy-Based VPN in Juniper Networks Virtual Labs

RELATED DOCUMENTATION

AutoVPN on Hub-and-Spoke Devices | **1021**

Configure Policy-Based IPsec VPN with Certificates

IN THIS SECTION

This example shows how to configure, verify, and troubleshoot PKI. This topic includes the following sections:

Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.4 or later
- Juniper Networks security devices

Before you begin:

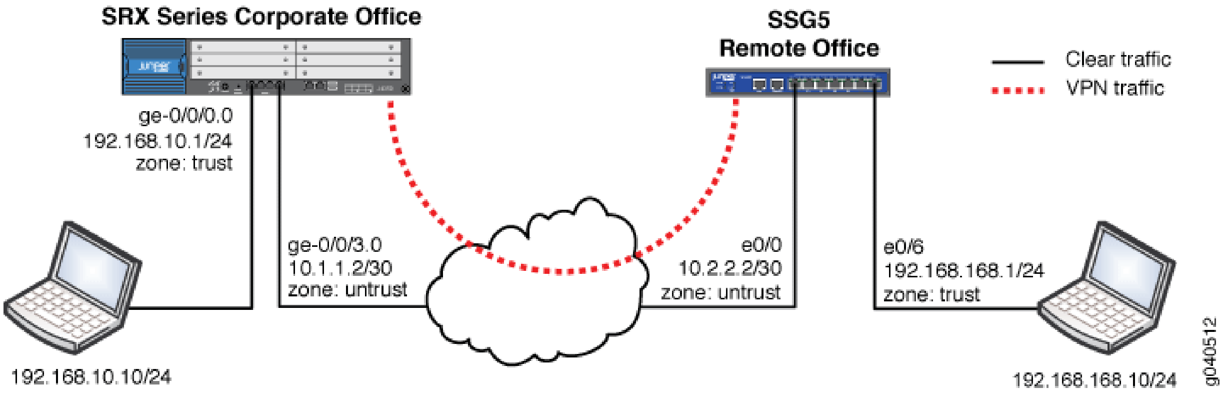
- Ensure that the internal LAN interface of the SRX Series Firewall is ge-0/0/0 in zone trust and has a private IP subnet.
- Ensure that the Internet interface of the device is ge-0/0/3 in zone untrust and has a public IP.
- Ensure that all traffic between the local and remote LANs is permitted, and traffic can be initiated from either side.
- Ensure that the SSG5 has been preconfigured correctly and loaded with a ready-to-use local certificate, CA certificate, and CRL.
- Ensure that the SSG5 device is configured to use the FQDN of ssg5.example.net (IKE ID).
- Ensure that PKI certificates with 1024-bit keys are used for the IKE negotiations on both sides.

- Ensure that the CA is a standalone CA at the domain example.com for both VPN peers.

Overview

Figure 27 on page 294 shows the network topology used for this example to configure a policy-based IPsec VPN to allow data to be securely transferred between a corporate office and a remote office.

Figure 27: Network Topology Diagram



The PKI administration is the same for both policy-based VPNs and route-based VPNs.

In this example, the VPN traffic is incoming on interface ge-0/0/0.0 with the next hop of 10.1.1.1. Thus the traffic is outgoing on interface ge-0/0/3.0. Any tunnel policy must consider incoming and outgoing interfaces.

Optionally, you can use a dynamic routing protocol such as OSPF (not described in this document). When processing the first packet of a new session, the device running Junos OS first performs a route lookup. The static route, which is also the default route, dictates the zone for the outgoing VPN traffic.

Many CAs use hostnames (for example, FQDN) to specify various elements of the PKI. Because the CDP is usually specified using a URL containing an FQDN, you must configure a DNS resolver on the device running Junos OS.

The certificate request can be generated by the following methods:

- Creating a CA profile to specify the CA settings
- Generating the PKCS10 certificate request

The PKCS10 certificate request process involves generating a public or private key pair and then generating the certificate request itself, using the key pair.

Take note of the following information about the CA profile:

- The CA profile defines the attributes of a certificate authority.
- Each CA profile is associated with a CA certificate. If a new or renewed CA certificate needs to be loaded without removing the older CA certificate, a new profile must be created. This profile can also be used for online fetching of the CRL.
- There can be multiple such profiles present in the system created for different users.

If you specify a CA administrator e-mail address to send the certificate request to, then the system composes an e-mail from the certificate request file and forwards it to the specified e-mail address. The e-mail status notification is sent to the administrator.

The certificate request can be sent to the CA through an out-of-band method.

The following options are available to generate the PKCS10 certificate request:

- `certificate-id` — Name of the local digital certificate and the public/private key pair. This ensures that the proper key pair is used for the certificate request and ultimately for the local certificate.

  Starting in Junos OS Release 19.1R1, a commit check is added to prevent user from adding `.`, `/`, `%`, and space in a certificate identifier while generating a local or remote certificates or a key pair.

- `subject` — Distinguished name format that contains the common name, department, company name, state, and country:

  - CN — Common name
  - OU — Department
  - O — Company name
  - L — Locality
  - ST — State
  - C — Country
  - CN — Phone
  - DC — Domain component

  You are not required to enter all subject name components. Note also that you can enter multiple values of each type.

- `domain-name` — FQDN. The FQDN provides the identity of the certificate owner for IKE negotiations and provides an alternative to the subject name.

- `filename (path | terminal)` — (Optional) Location where the certificate request should be placed, or the login terminal.

- `ip-address` — (Optional) IP address of the device.

- `email` — (Optional) E-mail address of the CA administrator.

  You must use a domain-name, an ip-address, or an e-mail address.

The generated certificate request is stored in a specified file location. A local copy of the certificate request is saved in the local certificate storage. If the administrator reissues this command, the certificate request is generated again.

The PKCS10 certificate request is stored in a specified file and location, from which you can download it and send it to the CA for enrollment. If you have not specified the filename or location, you can get PKCS10 certificate request details by using the `show security pki certificate-request certificate-id <id-name>` command in the CLI. You can copy the command output and paste it into a Web front end for the CA server or into an e-mail.

The PKCS10 certificate request is generated and stored on the system as a pending certificate or certificate request. An e-mail notification is sent to the administrator of the CA (in this example, certadmin@example.com).

A unique identity called certificate-ID is used to name the generated key pair. This ID is also used in certificate enrollment and request commands to get the right key pair. The generated key pair is saved in the certificate store in a file with the same name as the certificate-ID. The file size can be 1024 or 2048 bits.

A default (fallback) profile can be created if intermediate CAs are not preinstalled in the device. The default profile values are used in the absence of a specifically configured CA profile.

In the case of a CDP, the following order is followed:

- Per CA profile

- CDP embedded in CA certificate

- Default CA profile

We recommend using a specific CA profile instead of a default profile.

The administrator submits the certificate request to the CA. The CA administrator verifies the certificate request and generates a new certificate for the device. The administrator for the Juniper Networks device retrieves it, along with the CA certificate and CRL.

The process of retrieving the CA certificate, the device's new local certificate, and the CRL from the CA depends on the CA configuration and software vendor in use.

Junos OS supports the following CA vendors:

- Entrust
- Verisign
- Microsoft

Although other CA software services such as OpenSSL can be used to generate certificates, these certificates are not verified by Junos OS.

Configuration

IN THIS SECTION

PKI Basic Configuration

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure PKI:

1. Configure an IP address and protocol family on the Gigabit Ethernet interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet address 192.168.10.1/24
user@host# set ge-0/0/3 unit 0 family inet address 10.1.1.2/30
```

2. Configure a default route to the Internet next hop.

   ```
   [edit]
   user@host# set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
   ```

3. Set the system time and date.

   ```
   [edit]
   user@host# set system time-zone PST8PDT
   ```

   After the configuration is committed, verify the clock settings using the `show system uptime` command.

   ```
   user@host> show system uptime
   Current time: 2007-11-01 17:57:09 PDT
   System booted: 2007-11-01 14:36:38 PDT (03:20:31 ago)
   Protocols started: 2007-11-01 14:37:30 PDT (03:19:39 ago)
   Last configured: 2007-11-01 17:52:32 PDT (00:04:37 ago) by root
   5:57PM up 3:21, 4 users, load averages: 0.00, 0.00, 0.00
   ```

4. Set the NTP server address.

   ```
   user@host> set date ntp 130.126.24.24
   1 Nov 17:52:52 ntpdate[5204]: step time server 172.16.24.24 offset -0.220645 sec
   ```

5. Set the DNS configuration.

   ```
   [edit]
   user@host# set system name-server 172.31.2.1
   user@host# set system name-server 172.31.2.2
   ```

Configuring a CA Profile

Step-by-Step Procedure

1. Create a trusted CA profile.

   ```
   [edit]
   user@host# set security pki ca-profile ms-ca ca-identity example.com
   ```

2. Create a revocation check to specify a method for checking certificate revocation.

   Set the refresh interval, in hours, to specify the frequency in which to update the CRL. The default values are next-update time in CRL, or 1 week, if no next-update time is specified.

   ```
   [edit]
   user@host# set security pki ca-profile ms-ca revocation-check crl refresh-interval 48
   ```

   In the `revocation-check` configuration statement, you can use the `disable` option to disable the revocation check or select the `crl` option to configure the CRL attributes. You can select the `disable on-download-failure` option to allow the sessions matching the CA profile, when CRL download failed for a CA profile. The sessions will be allowed only if no old CRL is present in the same CA profile.

3. Specify the location (URL) to retrieve the CRL (HTTP or LDAP). By default, the URL is empty and uses CDP information embedded in the CA certificate.

   ```
   [edit]
   user@host# set security pki ca-profile ms-ca revocation-check crl url http://srv1.example.com/
   CertEnroll/EXAMPLE.crl
   ```

   Currently you can configure only one URL. Support for backup URL configuration is not available.

4. Specify an e-mail address to send the certificate request directly to a CA administrator.

   ```
   user@host# set security pki ca-profile ms-ca administrator email-address certadmin@example.com
   ```

5. Commit the configuration:

```
user@host# commit and-quit
commit complete
Exiting configuration mode
```

Generating a Public-Private Key Pair

Step-by-Step Procedure

When the CA profile is configured, the next step is to generate a key pair on the Juniper Networks device. To generate the private and public key pair:

1. Create a certificate key pair.

```
user@host> request security pki generate-key-pair certificate-id ms-cert size 1024
```

Results

After the public-private key pair is generated, the Juniper Networks device displays the following:

```
Generated key pair ms-cert, key size 1024 bits
```

Enrolling a Local Certificate

Step-by-Step Procedure

1. Generate a local digital certificate request in the PKCS-10 format. See "request security pki generate-certificate-request" on page 1736.

```
user@host> request security pki generate-certificate-request certificate-id ms-cert subject
"CN=john doe,CN=10.1.1.2,OU=sales,O=example, L=Sunnyvale,ST=CA,C=US" email user@example.net
filename ms-cert-req
Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIB3DCCAUUCAQAwbDERMA8GA1UEAxMIam9obiBkb2UxDjAMBgNVBAsTBXNhbGVz
MRkwFwYDVQQKExBKdW5pcGVyIE5ldHdvcmtzMRIwEAYDVQQHEwlTdW5ueXZhbGUx
```

```
CzAJBgNVBAgTAkNBMQswCQYDVQQGEwJVUzCBnzANBgkqhkiG9w0BAQEFAAOBjQAw
gYkCgYEA5EG6sgG/CTFzX6KC/hz6Czal0BxakUxfGxF7UWYWHaWFFYLqo6vXNO8r
OS5Yak7rWANAsMob3E2X/1adlQIRi4QFTjkBqGI+MTEDGnqFsJBqrB6oyqGtdcSU
u0qUivMvgKQVCx8hpx99J3EBTurfWL1pCNlBmZggNogb6MbwES0CAwEAAaAwMC4G
CSqGSIb3DQEJDjEhMB8wHQYDVR0RBBYwFIESInVzZXJAanVuaXBlci5uZXQiMA0G
CSqGSIb3DQEBBQUAA4GBAI6GhBaCsXk6/1lE2e5AakFFDhY7oqzHhgd1yMjiSUMV
djmf9JbDz2gM2UKpI+yKgtUjyCK/lV2ui57hpZMvnhAW4AmgwkOJg6mpR5rsxdLr
4/HHSHuEGOF17RHO6x0YwJ+KE1rYDRWj3Dtz447ynaLxcDF7buwd4IrMcRJJI9ws
-----END CERTIFICATE REQUEST-----
Fingerprint:
47:b0:e1:4c:be:52:f7:90:c1:56:13:4e:35:52:d8:8a:50:06:e6:c8 (sha1)
a9:a1:cd:f3:0d:06:21:f5:31:b0:6b:a8:65:1b:a9:87 (md5)
```

In the sample of the PKCS10 certificate, the request starts with and includes the BEGIN CERTIFICATE REQUEST line and ends with and includes the END CERTIFICATE REQUEST line. This portion can be copied and pasted to your CA for enrollment. Optionally, you can also offload the ms-cert-req file and send that to your CA.

2. Submit the certificate request to the CA, and retrieve the certificate.

Loading CA and Local Certificates

Step-by-Step Procedure

1. Load the local certificate, CA certificate, and CRL.

```
user@host> file copy ftp://192.168.10.10/certnew.cer certnew.cer /var/
tmp//...transferring.file.........crYdEC/100% of 1459 B 5864 kBps
user@host> file copy ftp:// 192.168.10.10/CA-certnew.cer CA-certnew.cer /var/
tmp//...transferring.file.........UKXUWu/100% of 1049 B 3607 kBps
user@host> file copy ftp:// 192.168.10.10/certcrl.crl certcrl.crl /var/
tmp//...transferring.file.........wpqnpA/100% of 401 B 1611 kBps
```

You can verify that all files have been uploaded by using the command `file list`.

2. Load the certificate into local storage from the specified external file.

You must also specify the certificate ID to keep the proper linkage with the private or public key pair. This step loads the certificate into the RAM cache storage of the PKI module, checks the associated private key, and verifies the signing operation.

```
user@host> request security pki local-certificate load certificate-id ms-cert filename
certnew.cer
Local certificate loaded successfully
```

3. Load the CA certificate from the specified external file.

   You must specify the CA profile to associate the CA certificate to the configured profile.

```
user@host> request security pki ca-certificate load ca-profile ms-ca filename CA-certnew.cer
Fingerprint:
1b:02:cc:cb:0f:d3:14:39:51:aa:0f:ff:52:d3:38:94:b7:11:86:30 (sha1)
90:60:53:c0:74:99:f5:da:53:d0:a0:f3:b0:23:ca:a3 (md5)
Do you want to load this CA certificate ? [yes,no] (no) yes
CA certificate for profile ms-ca loaded successfully
```

4. Load the CRL into the local storage.

   The maximum size of the CRL is 5 MB. You must specify the associated CA profile in the command.

```
user@host> request security pki crl load ca-profile ms-ca filename certcrl.crl
CRL for CA profile ms-ca loaded successfully
```

Results

Verify that all local certificates are loaded.

```
user@host> show security pki local-certificate certificate-id ms-cert detail Certificate
identifier: ms-cert
Certificate version: 3
Serial number: 3a01c5a0000000000011
Issuer:
Organization: Example, Organizational unit: example, Country: US, State:
CA, Locality: Sunnyvale,
Common name: LAB
Subject:
```

```
Organization: Example, Organizational unit: example, Country: US,
State: CA, Locality: Sunnyvale,
Common name: john doe
Alternate subject: "user@example.net", fqdn empty, ip empty
Validity:
Not before: 11- 2-2007 22:54
Not after: 11- 2-2008 23:04
Public key algorithm: rsaEncryption(1024 bits)
30:81:89:02:81:81:00:e4:41:ba:b2:01:bf:09:31:73:5f:a2:82:fe
1c:fa:0b:36:a5:d0:1c:5a:91:4c:5f:1b:11:7b:51:66:16:1d:a5:85
15:82:ea:a3:ab:d7:34:ef:2b:39:2e:58:6a:4e:eb:58:03:40:b0:ca
1b:dc:4d:97:ff:56:9d:95:02:11:8b:84:05:4e:39:01:a8:62:3e:31
31:03:1a:7a:85:b0:90:6a:ac:1e:a8:ca:a1:ad:75:c4:94:bb:4a:94
8a:f3:2f:80:a4:15:0b:1f:21:a7:1f:7d:27:71:01:4e:ea:df:58:bd
69:08:d9:41:99:98:20:36:88:1b:e8:c6:f0:11:2d:02:03:01:00:01
Signature algorithm: sha1WithRSAEncryption
Distribution CRL:
ldap:///CN=LAB,CN=LABSRV1,CN=CDP,CN=Public%20Key%20Services,CN=Services,
CN=Configuration,DC=domain,DC=com?certificateRevocationList?base?
objectclass=cRLDistributionPoint
http://labsrv1.domain.com/CertEnroll/LAB.crl
Fingerprint:
c9:6d:3d:3e:c9:3f:57:3c:92:e0:c4:31:fc:1c:93:61:b4:b1:2d:58 (sha1)
50:5d:16:89:c9:d3:ab:5a:f2:04:8b:94:5d:5f:65:bd (md5)
```

You can display the individual certificate details by specifying certificate-ID in the command line.

Verify all CA certificates or the CA certificates of an individual CA profile (specified).

```
user@host> show security pki ca-certificate ca-profile ms-ca detail
Certificate identifier: ms-ca
Certificate version: 3
Serial number: 44b033d1e5e158b44597d143bbfa8a13
Issuer:
Organization: Example, Organizational unit: example, Country: US, State:
CA, Locality: Sunnyvale,
Common name: example
Subject:
Organization: Example, Organizational unit: example, Country: US, State:
CA, Locality: Sunnyvale,
Common name: example
Validity:
```

```
Not before: 09-25-2007 20:32
Not after: 09-25-2012 20:41
Public key algorithm: rsaEncryption(1024 bits)
30:81:89:02:81:81:00:d1:9e:6f:f4:49:c8:13:74:c3:0b:49:a0:56
11:90:df:3c:af:56:29:58:94:40:74:2b:f8:3c:61:09:4e:1a:33:d0
8d:53:34:a4:ec:5b:e6:81:f5:a5:1d:69:cd:ea:32:1e:b3:f7:41:8e
7b:ab:9c:ee:19:9f:d2:46:42:b4:87:27:49:85:45:d9:72:f4:ae:72
27:b7:b3:be:f2:a7:4c:af:7a:8d:3e:f7:5b:35:cf:72:a5:e7:96:8e
30:e1:ba:03:4e:a2:1a:f2:1f:8c:ec:e0:14:77:4e:6a:e1:3b:d9:03
ad:de:db:55:6f:b8:6a:0e:36:81:e3:e9:3b:e5:c9:02:03:01:00:01
Signature algorithm: sha1WithRSAEncryption
Distribution CRL:
ldap:///CN=LAB,CN=LABSRV1,CN=CDP,CN=Public%20Key%20Services,CN=Services,
CN=Configuration,DC=domain,DC=com?certificateRevocationList?base?
objectclass=cRLDistributionPoint
http://srv1.domain.com/CertEnroll/LAB.crl
Use for key: CRL signing, Certificate signing, Non repudiation
Fingerprint:
1b:02:cc:cb:0f:d3:14:39:51:aa:0f:ff:52:d3:38:94:b7:11:86:30 (sha1)
90:60:53:c0:74:99:f5:da:53:d0:a0:f3:b0:23:ca:a3 (md5)
```

Verify all loaded CRLs or the CRLs of the specified individual CA profile.

```
user@host> show security pki crl ca-profile ms-ca detail
CA profile: ms-ca
CRL version: V00000001
CRL issuer: emailAddress = certadmin@example.net, C = US, ST = CA,
L = Sunnyvale, O = Example, OU = example, CN = example
Effective date: 10-30-2007 20:32
Next update: 11- 7-2007 08:52
```

Verify the certificate path for the local certificate and the CA certificate.

```
user@host> request security pki local-certificate verify certificate-id ms-cert
Local certificate ms-cert verification success
user@host> request security pki ca-certificate verify ca-profile ms-ca
CA certificate ms-ca verified successfully
```

Configuring the IPsec VPN with the Certificates

Step-by-Step Procedure

To configure the IPsec VPN with the certificate, refer to the network diagram shown in Figure 27 on page 294

1. Configure security zones and assign interfaces to the zones.

   In this example packets are incoming on `ge-0/0/0`, and the ingress zone is the trust zone.

   ```
   [edit security zones security-zone]
   user@host# set trust interfaces ge-0/0/0.0
   user@host# set untrust interfaces ge-0/0/3.0
   ```

2. Configure host-inbound services for each zone.

   Host-inbound services are for traffic destined for the Juniper Networks device. These settings include but are not limited to the FTP, HTTP, HTTPS, IKE, ping, rlogin, RSH, SNMP, SSH, Telnet, TFTP, and traceroute.

   ```
   [edit security zones security-zone]
   user@host# set trust host-inbound-traffic system-services all
   user@host# set untrust host-inbound-traffic system-services ike
   ```

3. Configure the address book entries for each zone.

   ```
   [edit security zones security-zone]
   user@host# set trust address-book address local-net 192.168.10.0/24
   user@host# set untrust address-book address remote-net 192.168.168.0/24
   ```

4. Configure the IKE (Phase 1) proposal to use RSA encryption.

   ```
   [edit security ike proposal rsa-prop1]
   user@host# set authentication-method rsa-signatures
   user@host# set encryption-algorithm 3des-cbc
   user@host# set authentication-algorithm sha1
   user@host# set dh-group group2
   ```

5. Configure an IKE policy.

The phase 1 exchange can take place in either main mode or aggressive mode.

```
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals rsa-prop1
user@host# set certificate local-certificate ms-cert
user@host# set certificate peer-certificate-type x509- signature
user@host# set certificate trusted-ca use-all
```

6. Configure an IKE gateway.

   In this example, the peer is identified by an FQDN (hostname). Therefore the gateway IKE ID should be the remote peer domain name. You must specify the correct external interface or peer ID to properly identify the IKE gateway during Phase 1 setup.

```
[edit security ike gateway ike-gate]
user@host# set external-interface ge-0/0/3.0
user@host# set ike-policy ike-policy1
user@host# set dynamic hostname ssg5.example.net
```

7. Configure the IPsec policy.

   This example uses the Standard proposal set, which includes `esp-group2-3des-sha1` and `esp-group2-aes128-sha1` proposals. However, a unique proposal can be created and then specified in the IPsec policy if needed.

```
[edit security ipsec policy vpn-policy1]
user@host# set proposal-set standard
user@host# set perfect-forward-secrecy keys group2
```

8. Configure the IPsec VPN with an IKE gateway and IPsec policy.

   In this example, the ike-vpn VPN name must be referenced in the tunnel policy to create a security association. Additionally, if required, an idle time and a proxy ID can be specified if they are different from the tunnel policy addresses.

```
[edit security ipsec vpn ike-vpn ike]
user@host# set gateway ike-gate
user@host# set ipsec-policy vpn-policy1
```

9. Configure bidirectional tunnel policies for VPN traffic.

   In this example, traffic from the host LAN to the remote office LAN requires a from-zone trust to-zone untrust tunnel policy. However, if a session needs to originate from the remote LAN to the host LAN, then a tunnel policy in the opposite direction from from-zone untrust to-zone trust is also required. When you specify the policy in the opposite direction as the pair-policy, the VPN becomes bidirectional. Note that in addition to the permit action, you also need to specify the IPsec profile to be used. Note that for tunnel policies, the action is always permit. In fact, if you are configuring a policy with the deny action, you will not see an option for specifying the tunnel.

   ```
   [edit security policies from-zone trust to-zone untrust]
   user@host# set policy tunnel-policy-out match source-address local-net
   user@host# set policy tunnel-policy-out match destination-address remote-net
   user@host# set policy tunnel-policy-out match application any
   user@host# set policy tunnel-policy-out then permit tunnel ipsec-vpn ike-vpn pair-policy
   tunnel-policy-in
   user@host# top edit security policies from-zone untrust to-zone trust
   user@host# set policy tunnel-policy-in match source-address remote-net
   user@host# set policy tunnel-policy-in match destination-address local-net
   user@host# set policy tunnel-policy-in match application any
   user@host# set policy tunnel-policy-in then permit tunnel ipsec-vpn ike-vpn pair-policy
   tunnel-policy-out
   ```

10. Configure a source NAT rule and a security policy for Internet traffic.

    The device uses the specified source-nat interface, and translates the source IP address and port for outgoing traffic, using the IP address of the egress interface as the source IP address and a random higher port for the source port. If required, more granular policies can be created to permit or deny certain traffic.

    ```
    [edit security nat source rule-set nat-out]
    user@host#set from zone trust
    user@host#set to zone untrust
    user@host#set rule interface-nat match source-address 192.168.10.0/24
    user@host#set rule interface-nat match destination-address 0.0.0.0/0
    user@host#set rule interface-nat then source-nat interface
    ```

    ```
    [edit security policies from-zone trust to-zone untrust]
    user@host# set policy any-permit match source-address any
    user@host# set policy any-permit match destination-address any
    ```

```
user@host# set policy any-permit match application any
user@host# set policy any-permit then permit
```

11. Move the tunnel policy above the any-permit policy.

    ```
    [edit security policies from-zone trust to-zone untrust]
    user@host# insert policy tunnel-policy-out before policy any-permit
    ```

    The security policy should be below the tunnel policy in the hierarchy because the policy list is read from top to bottom. If this policy were above the tunnel policy, then the traffic would always match this policy and would not continue to the next policy. Thus no user traffic would be encrypted.

12. Configure the tcp-mss setting for TCP traffic across the tunnel.

    TCP-MSS is negotiated as part of the TCP 3-way handshake. It limits the maximum size of a TCP segment to accommodate the MTU limits on a network. This is very important for VPN traffic because the IPsec encapsulation overhead along with the IP and frame overhead can cause the resulting ESP packet to exceed the MTU of the physical interface, causing fragmentation. Because fragmentation increases the bandwidth and device resources usage, and in general it should be avoided.

    The recommended value to use for tcp-mss is 1350 for most Ethernet-based networks with an MTU of 1500 or higher. This value might need to be altered if any device in the path has a lower value of MTU or if there is any added overhead such as PPP, Frame Relay, and so on. As a general rule, you might need to experiment with different tcp-mss values to obtain optimal performance.

    ```
    user@host# set security flow tcp-mss ipsec-vpn mss mss-value
    Example:
    [edit]
    user@host# set security flow tcp-mss ipsec-vpn mss 1350
    user@host# commit and-quit
    commit complete
    Exiting configuration mode
    ```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Confirming IKE Phase 1 Status

Purpose

Confirm the VPN status by checking any IKE Phase 1 security associations status.

PKI related to IPsec tunnels is formed during Phase 1 setup. Completion of Phase 1 indicates that PKI was successful.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations


Index Remote Address State Initiator cookie Responder cookie Mode
 2010.2.2.2 UP af4f78bc135e4365 48a35f853ee95d21 Main
```

Meaning

The output indicates that:

- The remote peer is 10.2.2.2 and the status is UP, which means the successful association of Phase 1 establishment.
- The remote peer IKE ID, IKE policy, and external interfaces are all correct.
- Index 20 is a unique value for each IKE security association. You can use this output details to get further details on each security association. See "Getting Details on Individual Security Associations" on page 310.

Incorrect output would indicate that:

- The remote peer status is Down.
- There are no IKE security associations .
- There are IKE policy parameters, such as the wrong mode type (Aggr or Main), PKI issues, or Phase 1 proposals (all must match on both peers). For more information, see "Troubleshooting IKE, PKI, and IPsec Issues" on page 316.
- External interface is invalid for receiving the IKE packets. Check the configurations for PKI-related issues, check the key management daemon (kmd) log for any other errors, or run trace options to find the mismatch. For more information, see "Troubleshooting IKE, PKI, and IPsec Issues" on page 316.

Getting Details on Individual Security Associations

Purpose

Get details on individual IKE.

Action

From operational mode, enter the **show security ike security-associations index 20 detail** command.

```
user@host> show security ike security-associations index 20 detail
IKE peer 10.2.2.2, Index 20,
Role: Responder, State: UP
Initiator cookie: af4f78bc135e4365, Responder cookie: 48a35f853ee95d21
Exchange type: Main, Authentication method: RSA-signatures
Local: 10.1.1.2:500, Remote: 10.2.2.2:500
Lifetime: Expires in 23282 seconds
```

```
Algorithms:
Authentication : sha1
Encryption : 3des-cbc
Pseudo random function: hmac-sha1
Traffic statistics:
Input bytes : 10249
Output bytes : 4249
Input packets: 10
Output packets: 9
Flags: Caller notification sent
IPsec security associations: 2 created, 1 deleted
Phase 2 negotiations in progress: 0
```

Meaning

The output displays the details of the individual IKE SAs such as role (initiator or responder), status, exchange type, authentication method, encryption algorithms, traffic statistics, Phase 2 negotiation status, and so on.

You can use the output data to:

- Know the role of the IKE SA. Troubleshooting is easier when the peer has the responder role.
- Get the traffic statistics to verify the traffic flow in both directions.
- Get the number of IPsec security associations created or in progress.
- Get the status of any completed Phase 2 negotiations.

Confirming IPsec Phase 2 Status

Purpose

View IPsec (Phase 2) security associations.

When IKE Phase 1 is confirmed, view the IPsec (Phase 2) security associations.

Action

From operational mode, enter the **show security ipsec security-associations** command.

```
user@host> show security ipsec security-associations
```

```
total configured sa: 2
ID Gateway Port Algorithm SPI Life:sec/kb Mon vsys
<2 10.2.2.2 500 ESP:3des/sha1 bce1c6e0 1676/ unlim - 0
>2 10.2.2.2 500 ESP:3des/sha1 1a24eab9 1676/ unlim - 0
```

Meaning

The output indicates that:

- There is a configured IPsec SA pair available . The port number 500 indicates that a standard IKE port is used. Otherwise, it is Network Address Translation-Traversal (NAT-T), 4500, or random high port.
- The security parameter index (SPI) is used for both directions. The lifetime or usage limits of the SA is expressed either in seconds or in kilobytes. In the output, 1676/ unlim indicates Phase 2 lifetime is set to expire in 1676 seconds and there is no specified lifetime size.
- The ID number shows the unique index value for each IPsec SA.
- A hyphen (-) in the Mon column indicates that VPN monitoring is not enabled for this SA.
- The virtual system (vsys) is zero, which is the default value.

Phase 2 lifetime can be different from the Phase 1 lifetime because Phase 2 is not dependent on Phase 1 after the VPN is up.

Displaying IPsec Security Association Details

Purpose

Display the individual IPsec SA details identified by the index number.

Action

From operational mode, enter the **show security ipsec security-associations index 2 detail** command.

```
user@host> show security ipsec security-associations index 2 detail
Virtual-system: Root
Local Gateway: 10.1.1.2, Remote Gateway: 10.2.2.2
Local Identity: ipv4_subnet(any:0,[0..7]=192.168.10.0/24)
Remote Identity: ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
DF-bit: clear
Policy-name: tunnel-policy-out
Direction: inbound, SPI: bce1c6e0, AUX-SPI: 0
```

```
Hard lifetime: Expires in 1667 seconds
Lifesize Remaining: Unlimited
Soft lifetime: Expires in 1093 seconds
Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Anti-replay service: enabled, Replay window size: 32
Direction: outbound, SPI: 1a24eab9, AUX-SPI: 0
Hard lifetime: Expires in 1667 seconds
Lifesize Remaining: Unlimited
Soft lifetime: Expires in 1093 seconds
Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Anti-replay service: enabled, Replay window size: 32
```

Meaning

The output displays the local Identity and the remote Identity.

Note that a proxy ID mismatch can cause Phase 2 completion to fail. The proxy ID is derived from the tunnel policy (for policy-based VPNs). The local address and remote address are derived from the address book entries, and the service is derived from the application configured for the policy.

If Phase 2 fails due to a proxy ID mismatch, verify which address book entries are configured in the policy and ensure that the correct addresses are sent. Also ensure that the ports are matching. Double-check the service to ensure that the ports match for the remote and local servers.

If multiple objects are configured in a tunnel policy for source address, destination address, or application, then the resulting proxy ID for that parameter is changed to zeroes.

For example, assume the following scenario for a tunnel policy:

- Local addresses of 192.168.10.0/24 and 10.10.20.0/24
- Remote address of 192.168.168.0/24
- Application as junos-http

The resulting proxy ID is local 0.0.0.0/0, remote 192.168.168.0/24, service 80.

The resulting proxy IDs can affect the interoperability if the remote peer is not configured for the second subnet. Also, if you are employing a third-party vendor’s application, you might have to manually enter the proxy ID to match.

If IPsec fails to complete, then check the kmd log or use the `set traceoptions` command. For more information, see "Troubleshooting IKE, PKI, and IPsec Issues" on page 316.

Checking IPsec SA Statistics

Purpose

Check statistics and errors for an IPsec SA.

For troubleshooting purpose, check the Encapsulating Security Payload/Authentication Header (ESP/AH) counters for any errors with a particular IPsec SA.

Action

From operational mode, enter the **show security ipsec statistics index 2** command.

```
user@host> show security ipsec statistics index 2
ESP Statistics:
Encrypted bytes: 674784
Decrypted bytes: 309276
Encrypted packets: 7029
Decrypted packets: 7029
AH Statistics:
Input bytes: 0
Output bytes: 0
Input packets: 0
Output packets: 0
Errors:
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```

Meaning

An error value of zero in the output indicates a normal condition.

We recommend running this command multiple times to observe any packet loss issues across a VPN. Output from this command also displays the statistics for encrypted and decrypted packet counters, error counters, and so on.

You must enable security flow trace options to investigate which ESP packets are experiencing errors and why. For more information, see "Troubleshooting IKE, PKI, and IPsec Issues" on page 316.

Testing Traffic Flow Across the VPN

Purpose

Test traffic flow across the VPN after Phase 1 and Phase 2 have completed successfully. You can test traffic flow by using the `ping` command. You can ping from local host to remote host. You can also initiate pings from the Juniper Networks device itself.

This example shows how to initiate a ping request from the Juniper Networks device to the remote host. Note that when pings are initiated from the Juniper Networks device, the source interface must be specified to ensure that the correct route lookup takes place and the appropriate zones are referenced in the policy lookup.

In this example, the ge-0/0/0.0 interface resides in the same security zone as the local host and must be specified in the ping request so that the policy lookup can be from zone trust to zone untrust.

Action

From operational mode, enter the **ping 192.168.168.10 interface ge-0/0/0 count 5** command.

```
user@host> ping 192.168.168.10 interface ge-0/0/0 count 5
PING 192.168.168.10 (192.168.168.10): 56 data bytes
64 bytes from 192.168.168.10: icmp_seq=0 ttl=127 time=8.287 ms
64 bytes from 192.168.168.10: icmp_seq=1 ttl=127 time=4.119 ms
64 bytes from 192.168.168.10: icmp_seq=2 ttl=127 time=5.399 ms
64 bytes from 192.168.168.10: icmp_seq=3 ttl=127 time=4.361 ms
64 bytes from 192.168.168.10: icmp_seq=4 ttl=127 time=5.137 ms
--- 192.168.168.10 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.119/5.461/8.287/1.490 ms
```

Confirming the Connectivity

Purpose

Confirm the connectivity between a remote host and a local host.

Action

From operational mode, enter the **ping 192.168.10.10 from ethernet0/6** command.

```
ssg5-> ping 192.168.10.10 from ethernet0/6

Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 192.168.10.10, timeout is 1 seconds from ethernet0/6
!!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=4/4/5 ms
```

Meaning

You can confirm end-to-end connectivity by using the `ping` command from the remote host to the local host. In this example, the command is initiated from the SSG5 device.

Failed end-to-end connectivity can indicate an issue with routing, policy, end host, or encryption/ decryption of the ESP packets. To verify the exact causes of the failure:

- Check IPsec statistics for details on errors as described in "Checking IPsec SA Statistics" on page 314.
- Confirm end host connectivity by using the `ping` command from a host on the same subnet as the end host. If the end host is reachable by other hosts, then you can assume that the issue is not with the end host.
- Enable security flow trace options for troubleshooting the routing-related and policy-related issues.

Troubleshooting IKE, PKI, and IPsec Issues

IN THIS SECTION

Troubleshoot IKE, PKI, and IPsec issues.

Basic Troubleshooting Steps

Problem

The basic troubleshooting steps are as follows:

1. Identifying and isolating the problem.

2. Debugging the problem.

The common approach of starting troubleshooting is with the lowest layer of the OSI layers and working your way up the OSI stack to confirm the layer in which the failure occurs.

Solution

Basic steps for troubleshooting IKE, PKI, and IPsec are as follows:

- Confirm the physical connectivity of the Internet link at the physical and data link levels.

- Confirm that the Juniper Networks device has connectivity to the Internet next hop and connectivity to the remote IKE peer.

- Confirm IKE Phase 1 completion.

- Confirm IKE Phase 2 completion if IKE Phase 1 completion is successful.

- Confirm the traffic flow across the VPN (if the VPN is up and active).

Junos OS includes the trace options feature. Using this feature, you can enable a trace option flag to write the data from the trace option to a log file, which can be predetermined or manually configured and stored in flash memory. These trace logs can be retained even after a system reboot. Check the available flash storage before implementing trace options.

You can enable the trace options feature in configuration mode and commit the configuration to use the trace options feature. Similarly to disable trace options, you must deactivate trace options in configuration mode and commit the configuration.

Checking the Free Disk Space on Your Device

Problem

Check the statistics on the free disk space in your device file systems.

Solution

From operational mode, enter the **show system storage** command.

```
user@host> show system storage
Filesystem Size Used Avail Capacity Mounted on
/dev/ad0s1a 213M 74M 137M 35% /
devfs 1.0K 1.0K 0B 100% /dev
devfs 1.0K 1.0K 0B 100% /dev/
/dev/md0 180M 180M 0B 100% /junos
/cf 213M 74M 137M 35% /junos/cf
devfs 1.0K 1.0K 0B 100% /junos/dev/
procfs 4.0K 4.0K 0B 100% /proc
/dev/bo0s1e 24M 13K 24M 0% /config
/dev/md1 168M 7.6M 147M 5% /mfs
/cf/var/jail 213M 74M 137M 35% /jail/var
```

The `/dev/ad0s1a` represents the onboard flash memory and is currently at 35 percent capacity.

Checking the Log Files to Verify Different Scenarios and Uploading Log Files to an FTP

Problem

View the log files to check security IKE debug messages, security flow debugs, and the state of logging to the syslog.

Solution

From operational mode, enter the **show log kmd**, **show log pkid**, **show log security-trace**, and **show log messages** commands.

```
user@host> show log kmd
user@host> show log pkid
user@host> show log security-trace
user@host> show log messages
```

You can view a list of all logs in the /var/log directory by using the `show log` command.

Log files can also be uploaded to an FTP server by using the `file copy` command.

```
(operational mode):
user@host> file copy path/filename dest-path/filename
Example:
```

```
user@host> file copy /var/log/kmd ftp://192.168.10.10/kmd.log
```

```
ftp://192.168.10.10/kmd.log 100% of 35 kB 12 MBps
```

Enabling IKE Trace Options to View Messages on IKE

Problem

To view success or failure messages for IKE or IPsec, you can view the kmd log by using the `show log kmd` command. Because the kmd log displays some general messages, it can be useful to obtain additional details by enabling IKE and PKI trace options.

Generally, it is best practice to troubleshoot the peer that has the responder role. You must obtain the trace output from the initiator and responder to understand the cause of a failure.

Configure IKE tracing options.

Solution

```
user@host> configure
Entering configuration mode

[edit]
user@host# edit security ike traceoptions
[edit security ike traceoptions]
```

```
user@host# set file ?
Possible completions:
<filename> Name of file in which to write trace information
files Maximum number of trace files (2..1000)
match Regular expression for lines to be logged
no-world-readable Don't allow any user to read the log file
size Maximum trace file size (10240..1073741824)
world-readable Allow any user to read the log file
```

```
[edit security ike traceoptions]
```

```
user@host# set flag ?
Possible completions:
all Trace everything
certificates Trace certificate events
database Trace security associations database events
general Trace general events
ike Trace IKE module processing
parse Trace configuration processing
policy-manager Trace policy manager processing
routing-socket Trace routing socket messages
timer Trace internal timer events
```

If you do not specify file names for the <filename> field, then all IKE trace options are written to the kmd log.

You must specify at least one flag option to write trace data to the log. For example:

- `file size` — Maximum size of each trace file, in bytes. For example, 1 million (1,000,000) can generate a maximum file size of 1 MB.

- `files` — Maximum number of trace files to be generated and stored in a flash memory device.

You must commit your configuration to start the trace.

Enabling PKI Trace Options to View Messages on IPsec

Problem

Enable PKI trace options to identify whether an IKE failure is related to the certificate or to a non-PKI issue.

Solution

```
[edit security pki traceoptions]
```

```
user@host# set file ?
Possible completions:
<filename> Name of file in which to write trace information
files Maximum number of trace files (2..1000)
match Regular expression for lines to be logged
no-world-readable Don't allow any user to read the log file
size Maximum trace file size (10240..1073741824)
world-readable Allow any user to read the log file
```

```
[edit security pki traceoptions]
```

```
user@host# set flag ?
Possible completions:
all Trace with all flags enabled
certificate-verification PKI certificate verification tracing
online-crl-check PKI online crl tracing
```

Setting up IKE and PKI Trace Options to Troubleshoot IKE Setup Issues with Certificates

Problem

Configure the recommended settings for IKE and PKI trace options.

The IKE and PKI trace options use the same parameters, but the default filename for all PKI-related traces is found in the pkid log.

Solution

```
user@host> configure
Entering configuration mode

[edit security ike traceoptions]
user@host# set file size 1m
user@host# set flag ike
user@host# set flag policy-manager
user@host# set flag routing-socket
user@host# set flag certificates

[edit security pki traceoptions]
user@host# set file size 1m
user@host# set flag all
user@host# commit and-quit
commit complete
Exiting configuration mode
```

Analyzing the Phase 1 Success Message

Problem

Understand the output of the `show log kmd` command when the IKE Phase 1 and Phase 2 conditions are successful.

Solution

```
Nov 7 11:52:14 Phase-1 [responder] done for local=ipv4(udp:500,[0..3]=
10.1.1.2) remote=fqdn(udp:500,[0..15]=ssg5.example.net)
```

```
Nov 7 11:52:14 Phase-2 [responder] done for
p1_local=ipv4(udp:500,[0..3]=10.1.1.2) p1_remote=fqdn(udp:500,[0..15]=ssg5.example.net)
p2_local=ipv4_subnet(any:0,[0..7]=192.168.10.0/24)
p2_remote=ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
```

The sample output indicates:

- `10.1.1.2`—Local address.
- `ssg5.example.net` —Remote peer (hostname with FQDN).
- `udp: 500`—NAT-T was not negotiated.
- `Phase 1 [responder] done`—Phase 1 status, along with the role (initiator or responder).
- `Phase 2 [responder] done`—Phase 1 status, along with the proxy ID information.

  You can also confirm the IPsec SA status by using the verification commands mentioned in "Confirming IKE Phase 1 Status" on page 309.

Analyzing the Phase 1 Failure Message (Proposal Mismatch)

Problem

Understanding the output of the `show log kmd` command, where the IKE Phase 1 condition is a failure, helps in determining the reason for the VPN not establishing Phase 1.

Solution

```
Nov 7 11:52:14 Phase-1 [responder] failed with error(No proposal chosen) for
local=unknown(any:0,[0..0]=) remote=fqdn(udp:500,[0..15]=ssg5.example.net)
Nov 7 11:52:14 10.1.1.2:500 (Responder) <-> 10.2.2.2:500 { 011359c9 ddef501d - 2216ed2a bfc50f5f
[-
1] / 0x00000000 } IP; Error = No proposal chosen (14)
```

The sample output indicates:

- `10.1.1.2`—Local address.
- `ssg5.example.net` —Remote peer (hostname with FQDN).
- `udp: 500`—NAT-T was not negotiated.
- `Phase-1 [responder] failed with error (No proposal chosen)`—Phase 1 failure because of proposal mismatch.

To resolve this issue, ensure that the parameters for the IKE gateway Phase 1 proposals on both the responder and the initiator match. Also confirm that a tunnel policy exists for the VPN.

Analyzing the Phase 1 Failure Message (Authentication Failure)

Problem

Understand the output of the `show log kmd` command when the IKE Phase 1 condition is a failure. This helps in determining the reason for the VPN not establishing Phase 1.

Solution

```
Nov 7 12:06:36 Unable to find phase-1 policy as remote peer:10.2.2.2 is not recognized.
Nov 7 12:06:36 Phase-1 [responder] failed with error(Authentication failed) for
local=ipv4(udp:500,[0..3]=10.1.1.2) remote=ipv4(any:0,[0..3]=10.2.2.2)
Nov 7 12:06:36 10.1.1.2:500 (Responder) <-> 10.2.2.2:500 { f725ca38 dad47583 - dab1ba4c ae26674b
[-
1] / 0x00000000 } IP; Error = Authentication failed (24)
```

The sample output indicates:

- `10.1.1.2`—Local address.
- `10.2.2.2`—Remote peer
- `Phase 1 [responder] failed with error (Authentication failed)`—Phase 1 failure due to the responder not recognizing the incoming request originating from a valid gateway peer. In the case of IKE with PKI certificates, this failure typically indicates that an incorrect IKE ID type was specified or entered.

To resolve this issue, confirm that the correct peer IKE ID type is specified on the local peer based on the following:

- How the remote peer certificate was generated
- Subject Alternative Name or DN information in the received remote peer certificate

Analyzing the Phase 1 Failure Message (Timeout Error)

Problem

Understand the output of the `show log kmd` command when the IKE Phase 1 condition is a failure.

Solution

```
Nov 7 13:52:39 Phase-1 [responder] failed with error(Timeout) for local=unknown(any:0,[0..0]=)
remote=ipv4(any:0,[0..3]=10.2.2.2)
```

The sample output indicates:

- `10.1.1.2`—Llocal address.
- `10.2.2.2`—Remote peer.
- `Phase 1 [responder] failed with error(Timeout)`—Phase 1 failure.

  This error indicates that either the IKE packet is lost enroute to the remote peer or there is a delay or no response from the remote peer.

Because this timeout error is the result of waiting on a response from the PKI daemon, you must review the PKI trace options output to see whether there is a problem with PKI.

Analyzing the Phase 2 Failure Message

Problem

Understand the output of the `show log kmd` command when the IKE Phase 2 condition is a failure.

Solution

```
Nov 7 11:52:14 Phase-1 [responder] done for local=ipv4(udp:500,[0..3]=
10.1.1.2) remote=fqdn(udp:500,[0..15]=ssg5.example.net)
Nov 7 11:52:14 Failed to match the peer proxy ids
p2_remote=ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
p2_local=ipv4_subnet(any:0,[0..7]=10.10.20.0/24) for the remote peer:ipv4(udp:500,
[0..3]=10.2.2.2)
Nov 7 11:52:14 KMD_PM_P2_POLICY_LOOKUP_FAILURE: Policy lookup for Phase-2 [responder] failed for
p1_local=ipv4(udp:500,[0..3]=10.1.1.2) p1_remote=ipv4(udp:500,[0..3]=10.2.2.2)
p2_local=ipv4_subnet(any:0,[0..7]=10.10.20.0/24)
p2_remote=ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
Nov 7 11:52:14 10.1.1.2:500 (Responder) <-> 10.2.2.2:500 { 41f638eb cc22bbfe - 43fd0e85 b4f619d5
[0]
/ 0xc77fafcf } QM; Error = No proposal chosen (14)
```

The sample output indicates:

- `10.1.1.2`—Local address.
- `ssg5.example.net` —Remote peer (IKE ID type hostname with FQDN).
- `Phase 1 [responder] done`—Phase 1 success.
- `Failed to match the peer proxy ids`—The Incorrect proxy IDs are received. In the previous sample, the two proxy IDs received are 192.168.168.0/24 (remote) and 10.10.20.0/24 (local) (for service=any). Based on the configuration given in this example, the expected local address is 192.168.10.0/24. This shows that there is a mismatch of configurations on the local peer, resulting in the failure of proxy ID match.

  To resolve this issue, correct the address book entry or configure the proxy ID on either peer so that it matches the other peer.

  The output also indicates the reason for failure is `No proposal chosen`. However in this case you also see the message `Failed to match the peer proxy ids`.

Analyzing the Phase 2 Failure Message

Problem

Understand the output of the `show log kmd` command when the IKE Phase 2 condition is a failure.

Solution

```
Nov 7 11:52:14 Phase-1 [responder] done for local=ipv4(udp:500,[0..3]=
10.1.1.2) remote=fqdn(udp:500,[0..15]=ssg5.example.net)
Nov 7 11:52:14 10.1.1.2:500 (Responder) <-> 10.2.2.2:500 { cd9dff36 4888d398 - 6b0d3933 f0bc8e26
[0]
/ 0x1747248b } QM; Error = No proposal chosen (14)
```

The sample output indicates:

- `10.1.1.2` —Local address.
- `fqdn(udp:500,[0..15]=ssg5.example.net`—Remote peer.
- `Phase 1 [responder] done`—Phase 1 success.
- `Error = No proposal chosen`—No proposal was chosen during Phase 2. This issue is due to proposal mismatch between the two peers.

  To resolve this issue, confirm that the Phase 2 proposals match on both peers.

Troubleshooting Common Problems Related to IKE and PKI

Problem

Troubleshoot common problems related to IKE and PKI.

Enabling the trace options feature helps you to gather more information on the debugging issues than is obtainable from the normal log entries. You can use the trace options log to understand the reasons for IKE or PKI failures.

Solution

Methods for troubleshooting the IKE -and-PKI-related issues:

- Ensure that the clock, date, time zone, and daylight savings settings are correct. Use NTP to keep the clock accurate.
- Ensure that you use a two-letter country code in the "C=" (country) field of the DN.

  For example: use “US” and not “USA” or “United States.” Some CAs require that the country field of the DN be populated, allowing you to enter the country code value only with a two-letter value.
- Ensure that if a peer certificate is using multiple OU=or CN= fields, you are using the distinguished name with container method (the sequence must be maintained and is case- sensitive).
- If the certificate is not valid yet, check the system clock and, if required, adjust the system time zone or just add a day in the clock for a quick test.
- Ensure that a matching IKE ID type and value are configured.
- PKI can fail due to a revocation check failure. To confirm this, temporarily disable revocation checking and see whether IKE Phase 1 is able to complete.

  To disable revocation checking, use the following command in configure mode:

  ```
  set security pki ca-profile <ca-profile> revocation-check disable
  ```

RELATED DOCUMENTATION

IPsec Overview | 20

PKI Components In Junos OS | 33

Configure IPsec VPN with OCSP for Certificate Revocation Status

IN THIS SECTION

This example shows how to improve security by configuring two peers using the Online Certificate Status Protocol (OCSP) to check the revocation status of the certificates used in Phase 1 negotiations for the IPsec VPN tunnel.

Requirements

On each device:

- Obtain and enroll a local certificate. This can be done either manually or by using the Simple Certificate Enrollment Protocol (SCEP).
- Optionally, enable automatic renewal of the local certificate.
- Configure security policies to permit traffic to and from the peer device.

Overview

IN THIS SECTION

On both peers, a certificate authority (CA) profile OCSP-ROOT is configured with the following options:

- CA name is OCSP-ROOT.
- Enrollment URL is http://10.1.1.1:8080/scep/OCSP-ROOT/. This is the URL where SCEP requests to the CA are sent.
- The URL for the OCSP server is http://10.157.88.56:8210/OCSP-ROOT/.
- OCSP is used first to check the certificate revocation status. If there is no response from the OCSP server, then the certificate revocation list (CRL) is used to check the status. The CRL URL is http://10.1.1.1:8080/crl-as-der/currentcrl-45.crlid=45.
- The CA certificate received in an OCSP response is not checked for certificate revocation. Certificates received in an OCSP response generally have shorter lifetimes and a revocation check is not required.

Table 32 on page 329 shows the Phase 1 options used in this example.

Table 32: Phase 1 Options for OCSP Configuration Example

Option	Peer A	Peer B
IKE proposal	ike_prop	ike_prop
Authentication method	RSA signatures	RSA signatures
DH group	group2	group2
Authentication algorithm	SHA 1	SHA 1
Encryption algorithm	3DES CBC	3DES CBC
IKE policy	ike_policy	ike_policy
Mode	aggressive	aggressive
Proposal	ike_prop	ike_prop
Certificate	local-certificate localcert1	local-certificate localcert1

Table 32: Phase 1 Options for OCSP Configuration Example *(Continued)*

Option	Peer A	Peer B
IKE gateway	jsr_gateway	jsr_gateway
Policy	ike_policy	ike_policy
Gateway address	198.51.100.50	192.0.2.50
Remote identity	localcert11.example.net	-
Local identity	-	localcert11.example.net
External interface	reth1	ge-0/0/2.0
Version	v2	v2

Table 33 on page 330 shows the Phase 2 options used in this example.

Table 33: Phase 2 Options for OCSP Configuration Example

Option	Peer A	Peer B
IPsec proposal	ipsec_prop	ipsec_prop
Protocol	ESP	ESP
Authentication algorithm	HMAC SHA1-96	HMAC SHA1-96
Encryption algorithm	3DES CBC	3DES CBC
Lifetime seconds	1200	1200
Lifetime kilobytes	150,000	150,000

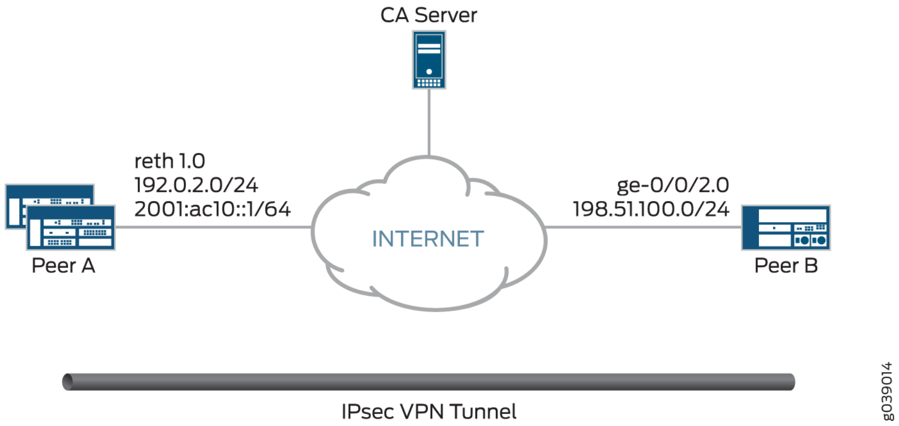
Table 33: Phase 2 Options for OCSP Configuration Example *(Continued)*

Option	Peer A	Peer B
IPsec policy	ipsec_policy	ipsec_policy
PFC keys	group2	group2
Proposal	ipsec_prop	ipsec_prop
VPN	test_vpn	test_vpn
Bind interface	st0.1	st0.1
IKE gateway	jsr_gateway	jsr_gateway
Policy	ipsec_policy	ipsec_policy
Establish tunnels	-	immediately

Topology

Figure 28 on page 332 shows the peer devices that are configured in this example.

Figure 28: OCSP Configuration Example



Configuration

IN THIS SECTION

Configuring Peer A

CLI Quick Configuration

To quickly configure VPN peer A to use OCSP, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/3 gigether-options redundant-parent reth1
set interfaces ge-9/0/3 gigether-options redundant-parent reth1
set interfaces lo0 unit 0 family inet address 172.16.1.100/24
set interfaces lo0 redundant-pseudo-interface-options redundancy-group 1
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 192.0.2.50/24
set interfaces st0 unit 1 family inet address 172.18.1.100/24
set security pki ca-profile OCSP-ROOT ca-identity OCSP-ROOT
set security pki ca-profile OCSP-ROOT enrollment url http://10.1.1.1:8080/scep/OCSP-ROOT/
set security pki ca-profile OCSP-ROOT revocation-check ocsp url http://10.157.88.56:8210/OCSP-
ROOT/
set security pki ca-profile OCSP-ROOT revocation-check use-ocsp
set security pki ca-profile OCSP-ROOT revocation-check ocsp disable-responder-revocation-check
set security pki ca-profile OCSP-ROOT revocation-check ocsp connection-failure fallback-crl
set security pki ca-profile OCSP-ROOT revocation-check crl url http://10.1.1.1:8080/crl-as-der/
currentcrl-45.crlid=45
set security ike proposal ike_prop authentication-method rsa-signatures
set security ike proposal ike_prop dh-group group2
set security ike proposal ike_prop authentication-algorithm sha1
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_policy mode aggressive
set security ike policy ike_policy proposals ike_prop
set security ike policy ike_policy certificate local-certificate localcert1
set security ike gateway jsr_gateway ike-policy ike_policy
set security ike gateway jsr_gateway address 198.51.100.50
set security ike gateway jsr_gateway remote-identity hostname localcert11.example.net
set security ike gateway jsr_gateway external-interface reth1
set security ike gateway jsr_gateway version v2-only
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec proposal ipsec_prop lifetime-seconds 1200
set security ipsec proposal ipsec_prop lifetime-kilobytes 150000
set security ipsec policy ipsec_policy perfect-forward-secrecy keys group2
set security ipsec policy ipsec_policy proposals ipsec_prop
set security ipsec vpn test_vpn bind-interface st0.1
set security ipsec vpn test_vpn ike gateway jsr_gateway
set security ipsec vpn test_vpn ike ipsec-policy ipsec_policy
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure VPN peer A to use OCSP:

1. Configure interfaces.

```
[edit interfaces]
set ge-0/0/3 gigether-options redundant-parent reth1
set ge-9/0/3 gigether-options redundant-parent reth1
set lo0 unit 0 family inet address 172.16.1.100/24
set  lo0 redundant-pseudo-interface-options redundancy-group 1
set reth1 redundant-ether-options redundancy-group 1
set reth1 unit 0 family inet address 192.0.2.0/24
set st0 unit 1 family inet address 172.18.1.100/24
```

2. Configure the CA profile.

```
[edit security pki ca-profile OCSP-ROOT]
set ca-identity OCSP-ROOT
set enrollment url http://10.1.1.1:8080/scep/OCSP-ROOT/
set revocation-check ocsp url http://10.157.88.56:8210/OCSP-ROOT/
set  revocation-check use-ocsp
set revocation-check ocsp disable-responder-revocation-check
set revocation-check ocsp connection-failure fallback-crl
set revocation-check crl url http://10.1.1.1:8080/crl-as-der/currentcrl-45.crlid=45
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike_prop]
set authentication-method rsa-signatures
set dh-group group2
set authentication-algorithm sha1
set encryption-algorithm 3des-cbc

[edit security ike policy ike_policy]
set mode aggressive
set proposals ike_prop
set certificate local-certificate localcert1

[edit security ike gateway jsr_gateway]
set ike-policy ike_policy
set address 198.51.100.50
set remote-identity hostname localcert11.example.net
set external-interface reth1
set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec_prop]
set protocol esp
set authentication-algorithm hmac-sha1-96
set encryption-algorithm 3des-cbc
set lifetime-seconds 1200
set lifetime-kilobytes 150000

[edit security ipsec policy ipsec_policy]
set perfect-forward-secrecy keys group2
set proposals ipsec_prop

[edit security ipsec vpn test_vpn]
set bind-interface st0.1
set ike gateway jsr_gateway
set ike ipsec-policy ipsec_policy
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security pki ca-profile OCSP-ROOT`, `show security ike`, and `show security ipsec` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/3 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-9/0/3 {
    gigether-options {
        redundant-parent reth1;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 172.16.1.100/24;
        }
    }
    redundant-pseudo-interface-options {
        redundancy-group 1;
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.0.2.0/24;
        }
    }
}
st0 {
    unit 1 {
        family inet {
            address 172.18.1.100/24;
        }
    }
}
[edit]
user@host# show security pki ca-profile OCSP-ROOT
ca-identity OCSP-ROOT;
enrollment {
    url http://10.1.1.1:8080/scep/OCSP-ROOT/;
}
revocation-check {
    crl {
        url http://10.1.1.1:8080/crl-as-der/currentcrl-45.crlid=45;
    }
    ocsp {
        disable-responder-revocation-check;
        url http://10.157.88.56:8210/OCSP-ROOT/;
    }
    use-ocsp;
}
[edit]
```

```
user@host# show security ike
proposal ike_prop {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
}
policy ike_policy {
    mode aggressive;
    proposals ike_prop;
    certificate {
        local-certificate localcert1;
    }
}
gateway jsr_gateway {
    ike-policy ike_policy;
    address 10.10.2.50;
    remote-identity hostname localcert11.example.net;
    external-interface reth1;
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 1200;
    lifetime-kilobytes 150000;
}
policy ipsec_policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec_prop;
}
vpn test_vpn {
    bind-interface st0.1;
    ike {
        gateway jsr_gateway;
        ipsec-policy ipsec_policy;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Peer B

CLI Quick Configuration

To quickly configure VPN peer B to use OCSP, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [`edit`] hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/2 unit 0 family inet address 198.51.100.0/24
set interfaces lo0 unit 0 family inet address 172.17.1.100/24
set interfaces st0 unit 1 family inet address 172.18.1.1/24
set security pki ca-profile OCSP-ROOT ca-identity OCSP-ROOT
set security pki ca-profile OCSP-ROOT enrollment url http://10.1.1.1:8080/scep/OCSP-ROOT/
set security pki ca-profile OCSP-ROOT revocation-check ocsp url http://10.157.88.56:8210/OCSP-
ROOT/
set security pki ca-profile OCSP-ROOT revocation-check use-ocsp
set security pki ca-profile OCSP-ROOT revocation-check ocsp disable-responder-revocation-check
set security pki ca-profile OCSP-ROOT revocation-check ocsp connection-failure fallback-crl
set security pki ca-profile OCSP-ROOT revocation-check crl url http://10.1.1.1:8080/crl-as-der/
currentcrl-45.crlid=45
set security ike proposal ike_prop authentication-method rsa-signatures
set security ike proposal ike_prop dh-group group2
set security ike proposal ike_prop authentication-algorithm sha1
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_policy mode aggressive
set security ike policy ike_policy proposals ike_prop
set security ike policy ike_policy certificate local-certificate localcert11
set security ike gateway jsr_gateway ike-policy ike_policy
set security ike gateway jsr_gateway address 192.0.2.50
set security ike gateway jsr_gateway local-identity hostname localcert11.example.net
set security ike gateway jsr_gateway external-interface ge-0/0/2.0
set security ike gateway jsr_gateway version v2-only
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec proposal ipsec_prop lifetime-seconds 1200
set security ipsec proposal ipsec_prop lifetime-kilobytes 150000
set security ipsec policy ipsec_policy perfect-forward-secrecy keys group2
set security ipsec policy ipsec_policy proposals ipsec_prop
set security ipsec vpn test_vpn bind-interface st0.1
set security ipsec vpn test_vpn ike gateway jsr_gateway
set security ipsec vpn test_vpn ike ipsec-policy ipsec_policy
set security ipsec vpn test_vpn establish-tunnels immediately
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure VPN peer B to use OCSP:

1. Configure interfaces.

```
[edit interfaces]
set ge-0/0/2 unit 0 family inet address 198.51.100.0/24
set lo0 unit 0 family inet address 172.17.1.100/24
set st0 unit 1 family inet address 172.18.1.1/24
```

2. Configure the CA profile.

```
[edit security pki ca-profile OCSP-ROOT]
set ca-identity OCSP-ROOT
set enrollment url http://10.1.1.1:8080/scep/OCSP-ROOT/
set revocation-check ocsp url http://10.157.88.56:8210/OCSP-ROOT/
set  revocation-check use-ocsp
set revocation-check ocsp disable-responder-revocation-check
set revocation-check ocsp connection-failure fallback-crl
set revocation-check crl url http://10.1.1.1:8080/crl-as-der/currentcrl-45.crlid=45
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike_prop]
set authentication-method rsa-signatures
set dh-group group2
set authentication-algorithm sha1
set encryption-algorithm 3des-cbc

[edit security ike policy ike_policy]
set mode aggressive
set proposals ike_prop
set certificate local-certificate localcert1

[edit security ike gateway jsr_gateway]
set ike-policy ike_policy
set address 192.0.2.50
set local-identity hostname localcert11.example.net
set external-interface ge-0/0/2.0
set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec_prop]
set protocol esp
set authentication-algorithm hmac-sha1-96
set encryption-algorithm 3des-cbc
set lifetime-seconds 1200
set lifetime-kilobytes 150000

[edit security ipsec policy ipsec_policy]
set perfect-forward-secrecy keys group2
set proposals ipsec_prop

[edit security ipsec vpn test_vpn]
set bind-interface st0.1
set ike gateway jsr_gateway
set ike ipsec-policy ipsec_policy
set establish-tunnels immediately
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security pki ca-profile OCSP-ROOT`, `show security ike`, and `show security ipsec` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/2 {
    unit 0 {
        family inet {
            address 198.51.100.0/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 172.17.1.100/24;
        }
    }
}
st0 {
    unit 1 {
        family inet {
            address 172.18.1.1/24;
        }
    }
}
[edit]
user@host# show security pki ca-profile OCSP-ROOT
ca-identity OCSP-ROOT;
enrollment {
    url http://10.1.1.1:8080/scep/OCSP-ROOT/;
}
revocation-check {
    crl {
        url http://10.1.1.1:8080/crl-as-der/currentcrl-45.crlid=45;
    }
    ocsp {
        disable-responder-revocation-check;
        url http://10.157.88.56:8210/OCSP-ROOT/;
    }
    use-ocsp;
}
[edit]
user@host# show security ike
proposal ike_prop {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
}
policy ike_policy {
    mode aggressive;
    proposals ike_prop;
    certificate {
        local-certificate localcert11;
    }
}
gateway jsr_gateway {
    ike-policy ike_policy;
    address 192.0.2.50;
    local-identity hostname localcert11.example.net;
    external-interface ge-0/0/2.0;
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 1200;
    lifetime-kilobytes 150000;
}
policy ipsec_policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec_prop;
}
vpn test_vpn {
    bind-interface st0.1;
    ike {
        gateway jsr_gateway;
        ipsec-policy ipsec_policy;
    }
    establish-tunnels immediately;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying CA Certificates

Purpose

Verify the validity of a CA certificate on each peer device.

Action

From operational mode, enter the `show security pki ca-certificate ca-profile OCSP-ROOT` or `show security pki ca-certificate ca-profile OCSP-ROOT detail` command.

```
user@host> show security pki ca-certificate ca-profile OCSP-ROOT
Certificate identifier: OCSP-ROOT
  Issued to: OCSP-ROOT, Issued by: C = US, O = example, CN = OCSP-ROOT
  Validity:
    Not before: 11-15-2013 22:26 UTC
    Not after: 11-14-2016 22:26 UTC
  Public key algorithm: rsaEncryption(2048 bits)

user@host> show security pki ca-certificate ca-profile OCSP-ROOT detail
Certificate identifier: OCSP-ROOT
  Certificate version: 3
  Serial number: 0000a17f
  Issuer:
    Organization: example, Country: US, Common name: OCSP-ROOT
  Subject:
    Organization: example, Country: US, Common name: OCSP-ROOT
  Subject string:
    C=US, O=example, CN=OCSP-ROOT
  Validity:
    Not before: 11-15-2013 22:26 UTC
    Not after: 11-14-2016 22:26 UTC
  Public key algorithm: rsaEncryption(2048 bits)
    30:82:01:0a:02:82:01:01:00:c6:38:e9:03:69:5e:45:d8:a3:ea:3d
    2e:e3:b8:3f:f0:5b:39:f0:b7:35:64:ed:60:a0:ba:89:28:63:29:e7
    27:82:47:c4:f6:41:53:c8:97:d7:1e:3c:ca:f0:a0:b9:09:0e:3d:f8
    76:5b:10:6f:b5:f8:ef:c5:e8:48:b9:fe:46:a3:c6:ba:b5:05:de:2d
    91:ce:20:12:8f:55:3c:a6:a4:99:bb:91:cf:05:5c:89:d3:a7:dc:a4
    d1:46:f2:dc:36:f3:f0:b5:fd:1d:18:f2:e6:33:d3:38:bb:44:8a:19
    ad:e0:b1:1a:15:c3:56:07:f9:2d:f6:19:f7:cd:80:cf:61:de:58:b8
    a3:f5:e0:d1:a3:3a:19:99:80:b0:63:03:1f:25:05:cc:b2:0c:cd:18
    ef:37:37:46:91:20:04:bc:a3:4a:44:a9:85:3b:50:33:76:45:d9:ba
    26:3a:3b:0d:ff:82:40:36:64:4e:ea:6a:d8:9b:06:ff:3f:e2:c4:a6
    76:ee:8b:58:56:a6:09:d3:4e:08:b0:64:60:75:f3:e2:06:91:64:73
    d2:78:e9:7a:cb:8c:57:0e:d1:9a:6d:3a:4a:9e:5b:d9:e4:a2:ef:31
    5d:2b:2b:53:ab:a1:ad:45:49:fd:a5:e0:8b:4e:0b:71:52:ca:6b:fa
    8b:0e:2c:7c:7b:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://10.1.1.1:8080/crl-as-der/currentcrl-45.crl?id=45
  Authority Information Access OCSP:
    http://10.1.1.1:8090/OCSP-ROOT/
  Use for key: CRL signing, Certificate signing, Key encipherment, Digital signature
  Fingerprint:
    ed:ce:ec:13:1a:d2:ab:0a:76:e5:26:6d:2c:29:5d:49:90:57:f9:41 (sha1)
    af:87:07:69:f0:3e:f7:c6:b8:2c:f8:df:0b:ae:b0:28 (md5)
```

In this example, IP addresses are used in the URLs in the CA profile configuration. If IP addresses are not used with CA-issued certificates or CA certificates, DNS must be configured in the device's configuration. DNS must be able to resolve the host in the distribution CRL and in the CA URL in the CA profile configuration. Additionally, you must have network reachability to the same host to receive revocation checks.

Meaning

The output shows the details and validity of CA certificate on each peer as follows:

- `C`—Country.
- `O`—Organization.
- `CN`—Common name.
- `Not before`—Begin date of validity.
- `Not after`—End date of validity.

Verifying Local Certificates

Purpose

Verify the validity of a local certificate on each peer device.

Action

From operational mode, enter the `show security pki local-certificate certificate-id localcert1 detail` command.

```
user@host> show security pki local-certificate certificate-id localcert1 detail
Certificate identifier: localcert1
  Certificate version: 3
  Serial number: 013e3f1d
  Issuer:
    Organization: example, Country: US, Common name: OCSP-ROOT
  Subject:
    Organization: example, Organizational unit: example, State: california1, Locality: 
sunnyvale1, Common name: localcert1, Domain component: domain_component1
  Subject string:
    DC=domain_component1, CN=localcert1, OU=example, O=example, L=sunnyvale1, ST=california1, 
C=us1
  Alternate subject: "localcert1@example.net", localcert1.example.net, 10.10.1.50
  Validity:
    Not before: 01-28-2014 22:23 UTC
    Not after: 03-29-2014 22:53 UTC
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:a6:df:c1:57:59:f8:4d:0f:c4:a8:96:25:97
    03:c4:a0:fb:df:d5:f3:d5:56:b6:5a:26:65:b8:1a:ec:be:f6:c6:5f
    b3:d7:d3:59:39:48:52:4a:e3:1b:e4:e0:6d:24:c3:c1:50:8c:55:3b
    c0:c1:29:a0:45:29:8e:ec:3e:52:2f:84:b3:e8:89:9a:0f:8b:7d:e8
    90:4b:c1:28:48:95:b3:aa:11:ab:b4:8c:a8:80:ce:90:07:2a:13:a2
    2f:84:44:92:3b:be:7d:39:5b:2f:9a:4c:7a:2f:2d:31:8b:12:6d:52
    34:7d:6b:e4:69:7e:f3:86:55:e2:89:31:98:c9:15:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://10.1.1.1:8080/crl-as-der/currentcrl-45.crl?id=45
  Authority Information Access OCSP:
    http://10.1.1.1/:8090/OCSP-ROOT/
  Fingerprint:
    00:c6:56:64:ad:e3:ce:8e:26:6b:df:17:1e:de:fc:14:a4:bb:8c:e4 (sha1)
    7f:43:c6:ed:e4:b3:7a:4f:9a:8c:0b:61:95:01:c9:52 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Meaning

The output shows the details and validity of a local certificate on each peer as follows:

- `DC`—Domain component.
- `CN`—Common name.
- `OU`—Organizational unit.
- `O`—Organization.
- `L`—Locality
- `ST`—State.
- `C`—Country.
- `Not before`—Begin date of validity.
- `Not after`—End date of validity.

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status on each peer device.

Action

From operational mode, enter the `show security ike security-associations` command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
6534660 UP     3e62e05abd6a703f  c552b238e8a26668  IKEv2          198.51.100.50
```

From operational mode, enter the `show security ike security-associations detail` command.

```
user@host> show security ike security-associations detail
IKE peer 198.51.100.50, Index 6534660, Gateway Name: jsr_gateway
  Role: Responder, State: UP
  Initiator cookie: 3e62e05abd6a703f, Responder cookie: c552b238e8a26668
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.0.2.50:500, Remote: 198.51.100.50:500
  Lifetime: Expires in 26906 seconds
  Peer ike-id: localcert11.example.net
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :                 2152
   Output bytes  :                 2097
   Input  packets:                    4
   Output packets:                    4
  Flags: IKE SA is created
  IPSec security associations: 4 created, 0 deleted
  Phase 2 negotiations in progress: 0

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 192.0.2.50:500, Remote: 198.51.100.50:500
    Local identity: 192.0.2.50
    Remote identity: localcert11.example.net
    Flags: IKE SA is created
```

Meaning

The `flags` field in the output shows that, IKE security association is created.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status on each peer device.

Action

From operational mode, enter the `show security ipsec security-associations` command.

```
user@host> show security ipsec security-associations
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131073 ESP:3des/sha1 9d1066e2 252/   150000 -   root 500   198.51.100.50
  >131073 ESP:3des/sha1 82079c2c 252/   150000 -   root 500   198.51.100.50
```

From operational mode, enter the `show security ipsec security-associations detail` command.

```
user@host> show security ipsec security-associations detail
  ID: 131073 Virtual-system: root, VPN Name: test_vpn
  Local Gateway: 192.0.2.50, Remote Gateway: 198.51.100.50
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
    DF-bit: clear
    Bind-interface: st0.1

  Port: 500, Nego#: 2, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Last Tunnel Down Reason: Delete payload received
    Direction: inbound, SPI: 9d1066e2, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 249 seconds
```

```
      Lifesize Remaining:  150000 kilobytes
      Soft lifetime: Expires in 10 seconds
      Mode: Tunnel(0 0), Type: dynamic, State: installed
      Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
      Anti-replay service: counter-based enabled, Replay window size: 64

      Direction: outbound, SPI: 82079c2c, AUX-SPI: 0
                                , VPN Monitoring: -
      Hard lifetime: Expires in 249 seconds
      Lifesize Remaining:  150000 kilobytes
      Soft lifetime: Expires in 10 seconds
      Mode: Tunnel(0 0), Type: dynamic, State: installed
      Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
      Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output shows the ipsec security associations details.

RELATED DOCUMENTATION

PKI Components In Junos OS | 33

IPv6 IPsec VPNs

IN THIS SECTION

Juniper Networks supports manual and autokey IKE with preshared keys configurations for IPv6 IPsec VPN.

VPN Feature Support for IPv6 Addresses

A route-based site-to-site VPN tunnel with a point-to-point secure tunnel interface can operate in IPv4-in-IPv4, IPv6-in-IPv6, IPv6-in-IPv4, or IPv4-in-IPv6 tunnel modes. IPv6 addresses can be in the outer IP header, which represents the tunnel endpoint, or in the inner IP header, which represents the final source and destination addresses for a packet.

Table 34 on page 350 defines the support for IPv6 addresses in VPN features.

Table 34: IPv6 Address Support in VPN Features

Feature	Supported	Exceptions
IKE and IPsec Support:		
IKEv1 and IKEv2	Yes	Unless specified, all supported features are applicable for IKEv1 and IKEv2.
Route-based VPN	Yes	–
Policy-based VPN	Yes	IPv6 policy-based VPNs are not supported on SRX Series Firewalls in chassis cluster configurations. IPv6 policy-based VPNs are only supported with IPv6-in-IPv6 tunnels on standalone SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.
Site-to-site VPN	Yes	Only one-to-one, site-to-site VPN is supported. Many-to-one, site-to-site VPN (NHTB) is not supported. NHTB configuration cannot be committed for tunnel modes other than IPv4-in-IPv4 tunnels.
Dynamic endpoint VPN	Yes	–
Dialup VPN	Yes	–

Table 34: IPv6 Address Support in VPN Features *(Continued)*

Feature	Supported	Exceptions
AutoVPN	Yes	AutoVPN networks that use secure tunnel interfaces in point-to-point mode support IPv6 addresses for traffic selectors and for IKE peers. AutoVPN in point-to-multipoint mode does not support IPv6 traffic.
Group VPN	No	–
Point-to-point tunnel interfaces	Yes	–
Point-to-multipoint tunnel interfaces	No	–
Hub-and-spoke scenario for site-to-site VPNs	Yes	–
Numbered and unnumbered tunnel interfaces	Yes	–
Unicast static and dynamic (RIP, OSPF, BGP) routing	Yes	–
Multicast dynamic routing (PIM)	No	–
Virtual router	Yes	–
Logical system	No	–
Automatic and manual SA and key management	Yes	–
Multiple SPUs	Yes	–

Table 34: IPv6 Address Support in VPN Features *(Continued)*

Feature	Supported	Exceptions
Chassis cluster	Yes	IPsec VPN with active-active mode is supported only on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices for route-based IPv6 tunnels. IPsec VPN with active-active mode is not supported on SRX5400, SRX5600, and SRX5800 devices.
Statistics, logs, per-tunnel debugging	Yes	–
SNMP MIB	Yes	–
Local address selection	Yes	When multiple addresses in the same address family are configured on a physical external interface to a VPN peer, we recommend that you also configure `local-address` at the `[edit security ike gateway` *`gateway-name`*`]` hierarchy level.
Loopback address termination	Yes	–
Xauth or modecfg over IPv6	No	–
SPC insert	Yes	–
ISSU	Yes	–
DNS name as IKE gateway address	Yes	As with IPv4 tunnels, peer gateway address changes in the DNS name are not supported with IPv6 tunnels.
Preshared key or certificate authentication	Yes	–

Table 34: IPv6 Address Support in VPN Features *(Continued)*

Feature	Supported	Exceptions
NAT-Traversal (NAT-T) for IPv4 IKE peers	Yes	NAT-T is supported only for IPv6-in-IPv4 and IPv4-in-IPv4 tunnel modes with IKEv1. IPv6-in-IPv6 and IPv4-in-IPv6 tunnel modes are not supported. IKEv2 is not supported for NAT-T. NAT-T from IPv6 to IPv4 or from IPv4 to IPv6 is not supported.
Dead peer detection (DPD) and DPD gateway failover	Yes	DPD gateway failover is only supported for different gateway addresses within the same family. Failover from an IPv6 gateway address to an IPv4 gateway address, or vice versa, is not supported.
Encryption sets, authentication algorithms, and DH groups supported in Junos OS Release 12.1X45-D10 release for SRX Series Firewalls.	Yes	–
Generic proposals and policies for IPv6 and IPv4	Yes	–
General IKE ID	Yes	–
ESP and AH transport modes	No	These modes are not supported for IPv4.
ESP and AH tunnel modes	Yes	AH tunnel mode with mutable extension headers and options is not supported.
Extended sequence number	No	–
Single proxy ID pairs	Yes	–
Multiple traffic selector pairs	Yes	Supported with IKEv1 only.
Lifetime of IKE or IPsec SA, in seconds	Yes	–

Table 34: IPv6 Address Support in VPN Features *(Continued)*

Feature	Supported	Exceptions
Lifetime of IKE SA, in kilobytes	Yes	–
VPN monitoring	No	Configuration with IPv6 tunnels cannot be committed.
DF bit	Yes	For IPv6-in-IPv6 tunnels, the DF bit is set only if configured at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level. `df-bit clear` is the default.
Dual-stack (parallel IPv4 and IPv6 tunnels) over a single physical interface	Yes	For route-based site-to-site VPNs. A single IPv4 tunnel can operate in both IPv4-in-IPv4 and IPv6-in-IPv4 tunnel modes and a single IPv6 tunnel can operate in both IPv4-in-IPv6 and IPv6-in-IPv6 tunnel modes.
IPv6 extension headers	Yes	IPv6 extension headers and IPv4 options for IKE and IPsec packets are accepted but are not processed. AH with mutable EHs and options is not supported.
Fragmentation and reassembly	Yes	–
VPN session affinity	Yes	–
Multicast traffic	No	–
Tunnel IP services (Screen, NAT, ALG, IPS, AppSecure)	Yes	–
Packet reordering for IPv6 fragments over tunnel	No	–
Bidirectional Forwarding Detection (BFD) over OSPFv3 routes on st0 interface	No	–

Table 34: IPv6 Address Support in VPN Features *(Continued)*

Feature	Supported	Exceptions
Neighbor Discovery Protocol (NDP) over st0 interfaces	No	–
PKI Support:		
PKI in virtual router	Yes	–
RSA signature authentication (512-, 1024-, 2048-, or 4096-bit key size)	Yes	–
DSA signature authentication (1024-, 2048-, or 4096-bit key size)	Yes	–
ECDSA signatures	Yes	–
Certificate chain authentication	No	–
Automatic or manual enrollment over IPv4	Yes	–
Automatic or manual revocation over IPv4	Yes	–
Automatic or manual enrollment over IPv6	No	–
Automatic or manual revocation over IPv6	No	–
IPv6 addresses within PKI certificate fields	No	–

SEE ALSO

Understanding VPN Tunnel Modes | 496

IPsec Overview | 20

Understanding IPv6 IKE and IPsec Packet Processing

IN THIS SECTION

This topic includes the following sections:

IPv6 IKE Packet Processing

Internet Key Exchange (IKE) is part of the IPsec suite of protocols. It automatically enables two tunnel endpoints to set up security associations (SAs) and negotiate secret keys with each other. There is no need to manually configure the security parameters. IKE also provides authentication for communicating peers.

IKE packet processing in IPv6 networks involves the following elements:

- Internet Security Association and Key Management Protocol (ISAKMP) Identification Payload

  ISAKMP identification payload is used to identify and authenticate the communicating IPv6 peers. Two ID types (ID_IPV6_ADDR and ID_IPV6_ADDR_SUBNET) are enabled for IPv6. The ID type indicates the type of identification to be used. The ID_IPV6_ADDR type specifies a single 16-octet IPv6 address. This ID type represents an IPv6 address. The ID_IPV6_ADDR_SUBNET type specifies a range of IPv6 addresses represented by two 16-octet values. This ID type represents an IPv6 network mask. Table 35 on page 356 lists the ID types and their assigned values in the identification payload.

  Table 35: ISAKMP ID Types and Their Values

ID Type	Value
RESERVED	0
ID_IPV4_ADDR	1
ID_FQDN	2

Table 35: ISAKMP ID Types and Their Values *(Continued)*

ID Type	Value
ID_USER_FQDN	3
ID_IPV4_ADDR_SUBNET	4
ID_IPV6_ADDR	5
ID_IPV6_ADDR_SUBNET	6
ID_IPV4_ADDR_RANGE	7
ID_IPV6_ADDR_RANGE	8
ID_DER_ASN1_DN	9
ID_DER_ASN1_GN	10
ID_KEY_ID	11
ID_LIST	12

The ID_IPV6_ADDR_RANGE type specifies a range of IPv6 addresses represented by two 16-octet values. The first octet value represents the starting IPv6 address and the second octet value represents the ending IPv6 address in the range. All IPv6 addresses falling between the first and last IPv6 addresses are considered to be part of the list.

Two ID types in ISAKMP identification payload (ID_IPV6_ADDR_RANGE and ID_IPV4_ADDR_RANGE) are not supported in this release.

- Proxy ID

  A proxy ID is used during Phase 2 of IKE negotiation. It is generated before an IPsec tunnel is established. A proxy ID identifies the SA to be used for the VPN. Two proxy IDs are generated—local and remote. The local proxy ID refers to the local IPv4 or IPv6 address/network and subnet mask. The remote proxy ID refers to the remote IPv4 or IPv6 address/network and subnet mask.

- Security Association

  An SA is an agreement between VPN participants to support secure communication. SAs are differentiated based on three parameters—security parameter index (SPI), destination IPv6 address, and security protocol (either AH or ESP). The SPI is a unique value assigned to an SA to help identify an SA among multiple SAs. In an IPv6 packet, the SA is identified from the destination address in the outer IPv6 header and the security protocol is identified from either the AH or the ESP header.

IPv6 IPsec Packet Processing

After IKE negotiations are completed and the two IKE gateways have established Phase 1 and Phase 2 SAs, IPv6 IPsec employs authentication and encryption technologies to secure the IPv6 packets. Because IPv6 addresses are 128 bits long compared to IPv4 addresses, which are 32-bits long, IPv6 IPsec packet processing requires more resources.

Packet reordering for IPv6 fragments over a tunnel is not supported.

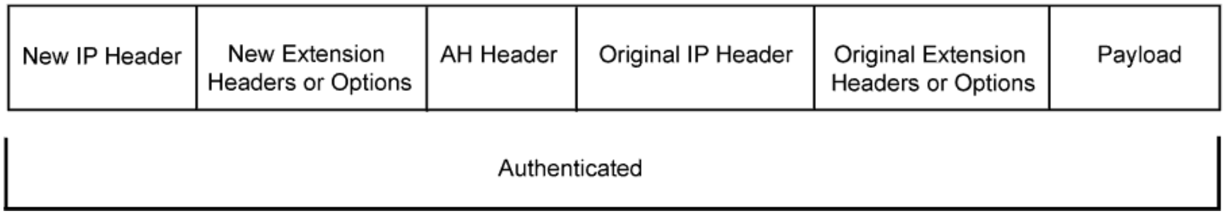
Devices with IPv6 addressing do not perform fragmentation. IPv6 hosts should either perform path MTU discovery or send packets smaller than the IPv6 minimum MTU size of 1280 bytes.

This topic includes the following sections:

AH Protocol in IPv6

The AH protocol provides data integrity and data authentication for IPv6 packets. IPv6 IPsec uses extension headers (for example, hop-by-hop and routing options) that must be arranged in a particular way in the IPv6 datagram. In AH tunnel mode, the AH header immediately follows the new outer IPv6 header similar to that in IPv4 AH tunnel mode. The extension headers are placed after the original inner header. Therefore, in AH tunnel mode, the entire packet is encapsulated by adding a new outer IPv6 header, followed by an authentication header, an inner header, extension headers, and the rest of the original datagram as shown in Figure 29 on page 358.

Figure 29: IPv6 AH Tunnel Mode


New IP Header
New Extension Headers or Options
AH Header
Original IP Header
Original Extension Headers or Options
Payload
Authenticated

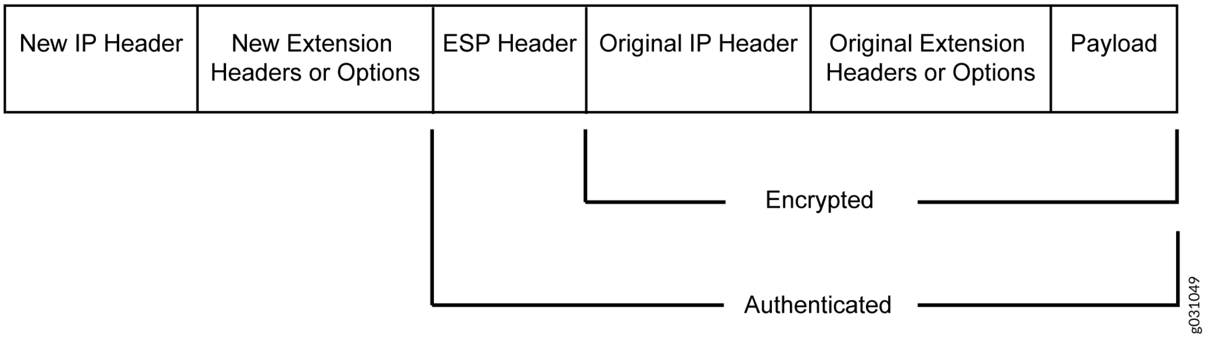
Unlike ESP, the AH authentication algorithm covers the outer header as well as any new extension headers and options.

AH tunnel mode on SRX Series Firewalls does not support IPv4 mutable options or IPv6 mutable extension headers. See Table 36 on page 359.

ESP Protocol in IPv6

ESP protocol provides both encryption and authentication for IPv6 packets. Because IPv6 IPsec uses extension headers (for example, hop-by-hop and routing options) in the IPv6 datagram, the most important difference between IPv6 ESP tunnel mode and IPv4 ESP tunnel mode is the placement of extension headers in the packet layout. In ESP tunnel mode, the ESP header immediately follows the new outer IPv6 header similar to that in IPv4 ESP tunnel mode. Therefore, in ESP tunnel mode, the entire packet is encapsulated by adding a new outer IPv6 header, followed by an ESP header, an inner header, extension headers, and the rest of the original datagram as shown in Figure 30 on page 359.

Figure 30: IPv6 ESP Tunnel Mode



IPv4 Options and IPv6 Extension Headers with AH and ESP

IPsec packets with IPv4 options or IPv6 extension headers can be received for decapsulation on SRX Series Firewalls. Table 36 on page 359 shows the IPv4 options or IPv6 extension headers that are supported with the ESP or AH protocol on SRX Series Firewalls. If an unsupported IPsec packet is received, ICV calculation fails and the packet is dropped.

Table 36: Support for IPv4 Options or IPv6 Extension Headers

Options or Extension Headers	SRX300, SRX320, SRX340, SRX345, and SRX550HM Devices	SRX5400, SRX5600, and SRX5800 Devices
ESP with IPv4 options	Supported	Supported

Table 36: Support for IPv4 Options or IPv6 Extension Headers *(Continued)*

Options or Extension Headers	SRX300, SRX320, SRX340, SRX345, and SRX550HM Devices	SRX5400, SRX5600, and SRX5800 Devices
ESP with IPv6 extension headers	Supported	Supported
AH with IPv4 immutable options	Supported	Supported
AH with IPv6 immutable extension headers	Supported	Supported
AH with IPv4 mutable options	Not supported	Not supported
AH with IPv6 mutable extension headers	Not supported	Not supported

Integrity Check Value Calculation in IPv6

The AH protocol verifies the integrity of the IPv6 packet by computing an Integrity Check Value (ICV) on the packet contents. ICV is usually built over an authentication algorithm such as MD5 or SHA-1. The IPv6 ICV calculations differ from that in IPv4 in terms of two header fields—mutable header and optional extension header.

You can calculate the AH ICV over the IPv6 header fields that are either immutable in transit or predictable in value upon arrival at the tunnel endpoints. You can also calculate the AH ICV over the AH header and the upper level protocol data (considered to be immutable in transit). You can calculate the ESP ICV over the entire IPv6 packet, excluding the new outer IPv6 header and the optional extension headers.

Unlike IPv4, IPv6 has a method for tagging options as mutable in transit. IPv6 optional extension headers contain a flag that indicates mutability. This flag determines the appropriate processing.

IPv4 mutable options and IPv6 extension headers are not supported with the AH protocol.

Header Construction in Tunnel Modes

In tunnel mode, the source and destination addresses of the outer IPv4 or IPv6 header represent the tunnel endpoints, while the source and destination addresses of the inner IPv4 or IPv6 header represent the final source and destination addresses. Table 37 on page 361 summarizes how the outer IPv6 header relates to the inner IPv6 or IPv4 header for IPv6-in-IPv6 or IPv4-in-IPv6 tunnel modes. In outer header fields, “Constructed” means that the value of the outer header field is constructed independently of the value in the inner header field.

Table 37: IPv6 Header Construction for IPv6-in-IPv6 and IPv4-in-IPv6 Tunnel Modes

Header Fields	Outer Header at Encapsulator	Inner Header at Decapsulator
version	6.	No change.
DS field	Copied from the inner header.	No change.
ECN field	Copied from the inner header.	Constructed.
flow label	0.	No change.
payload length	Constructed.	No change.
next header	AH, ESP, and routing header.	No change.
hop limit	64.	Decrement.
src address	Constructed.	No change.
dest address	Constructed.	No change.
Extension headers	Never copied.	No change.

Table 38 on page 361 summarizes how the outer IPv4 header relates to the inner IPv6 or IPv4 header for IPv6-in-IPv4 or IPv4-in-IPv4 tunnel modes. In outer header fields, “Constructed” means that the value of the outer header field is constructed independently of the value in the inner header field.

Table 38: IPv4 Header Construction for IPv6-in-IPv4 and IPv4-in-IPv4 Tunnel Modes

Header Fields	Outer Header	Inner Header
version	4.	No change.

Table 38: IPv4 Header Construction for IPv6-in-IPv4 and IPv4-in-IPv4 Tunnel Modes *(Continued)*

Header Fields	Outer Header	Inner Header
header length	Constructed.	No change.
DS field	Copied from the inner header.	No change.
ECN field	Copied from the inner header.	Constructed.
total length	Constructed.	No change.
ID	Constructed.	No change.
flags (DF, MF)	Constructed.	No change.
fragment offset	Constructed.	No change.
TTL	64.	Decrement.
protocol	AH, ESP	No change.
checksum	Constructed.	Constructed.
src address	Constructed.	No change.
dest address	Constructed.	No change.
options	Never copied.	No change.

For IPv6-in-IPv4 tunnel mode, the Don’t Fragment (DF) bit is cleared by default. If the `df-bit set` or `df-bit copy` options are configured at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level for the corresponding IPv4 VPN, the DF bit is set in the outer IPv4 header.

For IPv4-in-IPv4 tunnel mode, the DF bit in the outer IPv4 header is based on the `df-bit` option configured for the inner IPv4 header. If `df-bit` is not configured for the inner IPv4 header, the DF bit is cleared in the outer IPv4 header.

SEE ALSO

IPsec Overview | 20
IPv6 IPsec Configuration Overview | 363

IPv6 IPsec Configuration Overview

Juniper Networks supports manual and autokey IKE with preshared keys configurations for IPv6 IPsec VPN.

- AutoKey IKE VPN—In an autoKey IKE VPN configuration, the secret keys and SAs are automatically created using the autoKey IKE mechanism. To set up an IPv6 autoKey IKE VPN, two phases of negotiations are required—Phase 1 and Phase 2.

  - Phase 1—In this phase, the participants establish a secure channel for negotiating the IPsec SAs.

  - Phase 2—In this phase, the participants negotiate the IPsec SAs for authenticating and encrypting the IPv6 data packets.

  For more information on Phase 1 and Phase 2 negotiations, see "Internet Key Exchange" on page 10

SEE ALSO

IPsec VPN with Autokey IKE Configuration Overview | 190
Example: Configuring an IPv6 address as the Source Address for a CA Profile | 50

Example: Configuring an IPv6 IPsec Manual VPN

IN THIS SECTION

This example shows how to configure an IPv6 IPsec manual VPN.

Requirements

Before you begin:

- Understand how VPNs work. See "IPsec Overview" on page 20.
- Understand IPv6 IPsec packet processing. See "Understanding IPv6 IKE and IPsec Packet Processing" on page 356.

Overview

In a Manual VPN configuration, the secret keys are manually configured on the two IPsec endpoints.

In this example, you:

- Configure the authentication parameters for a VPN named vpn-sunnyvale.
- Configure the encryption parameters for vpn-sunnyvale.
- Specify the outgoing interface for the SA.
- Specify the IPv6 address of the peer.
- Define the IPsec protocol. Select the ESP protocol because the configuration includes both authentication and encryption.
- Configure a security parameter index (SPI).

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec vpn vpn-sunnyvale manual authentication algorithm hmac-md5-96 key ascii-text
“$ABC123”
set security ipsec vpn vpn-sunnyvale manual encryption algorithm 3des-cbc key ascii-text
“$ABC123”
set security ipsec vpn vpn-sunnyvale manual external-interface ge-0/0/14.0
set security ipsec vpn vpn-sunnyvale manual gateway 2001:db8:1212::1112
set security ipsec vpn vpn-sunnyvale manual protocol esp
set security ipsec vpn vpn-sunnyvale manual spi 12435
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure security algorithms:

1. Configure the authentication parameters.

   ```
   [edit security ipsec vpn vpn-sunnyvale manual]
   user@host# set authentication algorithm hmac-md5-96 key ascii-text “$ABC123”
   ```

2. Configure the encryption parameters.

   ```
   [edit security ipsec vpn vpn-sunnyvale manual]
   user@host# set encryption algorithm 3des-cbc key ascii-text “$ABC123”
   ```

3. Specify the outgoing interface for the SA.

   ```
   [edit security ipsec vpn vpn-sunnyvale manual]
   user@host# set external-interface ge-0/0/14.0
   ```

4. Specify the IPv6 address of the peer.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set gateway 2001:db8:1212::1112
```

5. Define the IPsec protocol.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set protocol esp
```

6. Configure an SPI.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set spi 12435
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec vpn vpn-sunnyvale` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
[user@host]show security ipsec vpn vpn-sunnyvale
manual {
gateway 2001:db8:1212::1112 ;
external-interface ge-0/0/14.0 ;
protocol esp ;
spi 12435 ;
authentication {
    algorithm hmac-md5-96 ;
    key ascii-text $ABC123” ;## SECRET DATA
}
    encryption {
        algorithm 3des-cbc ;
        key ascii-text $ABC123”; ## SECRET DATA
        }
    }
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform this task:

Verifying Security Algorithms

Purpose

Determine if security algorithms are applied or not.

Action

From operational mode, enter the `show security ipsec security-associations` command.

SEE ALSO

IPv6 IPsec Configuration Overview | 363

Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN

IN THIS SECTION

This example shows how to configure a policy-based IPv6 AutoKey IKE VPN to allow IPv6 data to be securely transferred between the branch office and the corporate office.

IPv6 policy-based VPNs are supported only on standalone SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Requirements

This example uses the following hardware:

- SRX300 device

Before you begin:

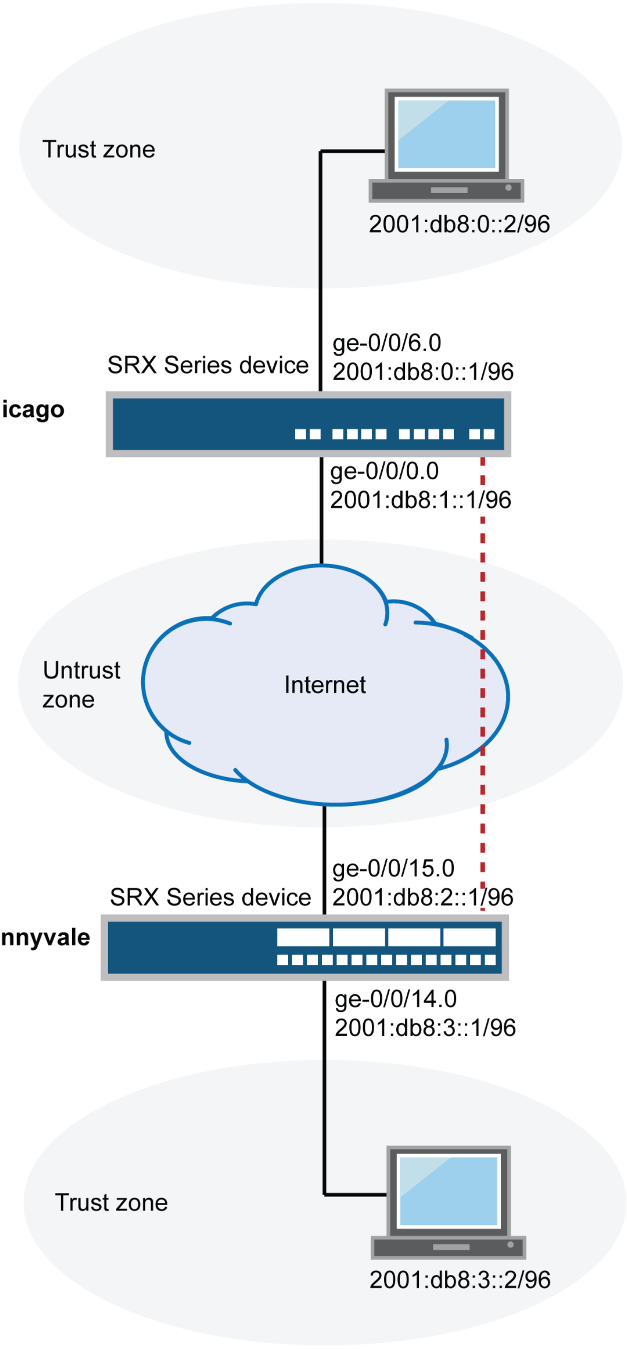
- Understand how VPNs work. See "IPsec Overview" on page 20.
- Understand IPv6 IKE and IPsec packet processing. See "Understanding IPv6 IKE and IPsec Packet Processing" on page 356.

Overview

In this example, you configure an IPv6 IKE policy-based VPN for a branch office in Chicago, Illinois, because you do not need to conserve tunnel resources or configure many security policies to filter traffic through the tunnel. Users in the Chicago office will use the VPN to connect to their corporate headquarters in Sunnyvale, California.

Figure 31 on page 369 shows an example of an IPv6 IKE policy-based VPN topology. In this topology, one SRX Series Firewall is located in Sunnyvale, and another SRX Series Firewall (this can be a second SRX Series Firewall or a third-party device) is located in Chicago.

Figure 31: IPv6 IKE Policy-Based VPN Topology



In this example, you configure interfaces, an IPv6 default route, security zones, and address books. Then you configure IKE Phase 1, IPsec Phase 2, a security policy, and TCP-MSS parameters. See Table 39 on page 370 through Table 43 on page 373.

Table 39: Interface, Security Zone, and Address Book Information

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/14.0	2001:db8:3::1/96
	ge-0/0/15.0	2001:db8:0:2::1/96
Security zones	Trust	• All system services are allowed. • The ge-0/0/14.0 interface is bound to this zone.
	Untrust	• IKE is the only allowed system service. • The ge-0/0/15.0 interface is bound to this zone.
Address book entries	Sunnyvale	• This address is for the Trust zone’s address book. • The address for this address book entry is 2001:db8:3::2/96.
	Chicago	• This address is for the Untrust zone’s address book. • The address for this address book entry is 2001:db8:0::2/96.

Table 40: IPv6 IKE Phase 1 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipv6-ike-phase1-proposal	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: aes-128-cbc
Policy	ipv6-ike-phase1-policy	• Mode: Aggressive • Proposal reference: ipv6-ike-phase1-proposal • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw-Chicago	• IKE policy reference: ipv6-ike-phase1-policy • External interface: ge-0/0/15.0 • Gateway address: 2001:db8:1::1/96

Table 41: IPv6 IPsec Phase 2 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipv6-ipsec-phase2-proposal	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: aes-128-cbc
Policy	ipv6-ipsec-phase2-policy	• Proposal reference: ipv6-ipsec-phase2-proposal • PFS: Diffie-Hellman group2

Table 41: IPv6 IPsec Phase 2 Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
VPN	ipv6-ike-vpn-chicago	• IKE gateway reference: gw-chicago • IPsec policy reference: ipv6-ipsec-phase2-policy

Table 42: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the Trust zone to the Untrust zone.	ipv6-vpn-tr-untr	• Match criteria: • source-address Sunnyvale • destination-address Chicago • application any • Permit action: tunnel ipsec-vpn ipv6-ike-vpn-chicago • Permit action: tunnel pair-policy ipv6-vpn-untr-tr
This security policy permits traffic from the Untrust zone to the Trust zone.	ipv6-vpn-untr-tr	• Match criteria: • source-address Chicago • destination-address Sunnyvale • application any • Permit action: tunnel ipsec-vpn ipv6-ike-vpn-chicago • Permit action: tunnel pair-policy ipv6-vpn-tr-untr

Table 42: Security Policy Configuration Parameters *(Continued)*

Purpose	Name	Configuration Parameters
This security policy permits all traffic from the Trust zone to the Untrust zone. You must put the ipv6-vpn-tr-untr policy before the permit-any security policy. Junos OS performs a security policy lookup starting at the top of the list. If the permit-any policy comes before the ipv6-vpn-tr-untr policy, all traffic from the Trust zone will match the permit-any policy and be permitted. Thus, no traffic will ever match the ipv6-vpn-tr-untr policy.	permit-any	• Match criteria: • source-address any • source-destination any • application any • Action: permit

Table 43: TCP-MSS Configuration Parameters

Purpose	Configuration Parameters
TCP-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the MTU limits on a network. This is especially important for VPN traffic, as the IPsec encapsulation overhead, along with the IP and frame overhead, can cause the resulting ESP packet to exceed the MTU of the physical interface, thus causing fragmentation. Fragmentation results in increased use of bandwidth and device resources. We recommend a value of 1350 as the starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.	MSS value: 1350

Configuration

IN THIS SECTION

Configuring Basic Network, Security Zone, and Address Book Information

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/14 unit 0 family inet6 address 2001:db8:3::1/96
set interfaces ge-0/0/15 unit 0 family inet6 address 2001:db8:2::1/96
set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
set security zones security-zone Untrust interfaces ge-0/0/15.0
set security zones security-zone Untrust host-inbound-traffic system-services ike
set security zones security-zone Trust interfaces ge-0/0/14.0
set security zones security-zone Trust host-inbound-traffic system-services all
set security address-book book1 address Sunnyvale 2001:db8:3::2/96
set security address-book book1 attach zone Trust
set security address-book book2 address Chicago 2001:db8:0::2/96
set security address-book book2 attach zone Untrust
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure basic network, security zone, and address book information:

1. Configure Ethernet interface information.

```
[edit]
user@host# set interfaces ge-0/0/14 unit 0 family inet6 address 2001:db8:3::1/96
user@host# set interfaces ge-0/0/15 unit 0 family inet6 address 2001:db8:2::1/96
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
```

3. Configure the Untrust security zone.

```
[edit]
user@host# edit security zones security-zone Untrust
```

4. Assign an interface to the Untrust security zone.

```
[edit security zones security-zone Untrust]
user@host# set interfaces ge-0/0/15.0
```

5. Specify allowed system services for the Untrust security zone.

```
[edit security zones security-zone Untrust]
user@host# set host-inbound-traffic system-services ike
```

6. Configure the Trust security zone.

```
[edit]
user@host# edit security zones security-zone Trust
```

7. Assign an interface to the Trust security zone.

```
[edit security zones security-zone Trust]
user@host# set interfaces ge-0/0/14.0
```

8. Specify allowed system services for the Trust security zone.

```
[edit security zones security-zone Trust]
user@host# set host-inbound-traffic system-services all
```

9. Create an address book and attach a zone to it.

```
[edit security address-book book1]
user@host# set address Sunnyvale 2001:db8:3::2/96
user@host# set attach zone Trust
```

10. Create another address book and attach a zone to it.

```
[edit security address-book book2]
user@host# set address Chicago 2001:db8:0::2/96
user@host# set attach zone Untrust
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security zones`, and `show security address-book` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/14 {
    unit 0 {
        family inet6 {
            address 2001:db8:3::1/96;
        }
    }
}
ge-0/0/15 {
    unit 0 {
        family inet6 {
            address 2001:db8:2::1/96;
        }
    }
}
```

```
[edit]
user@host# show routing-options
static {
    route 0.0.0.0/0 next-hop 10.1.1.1;
}
```

```
[edit]
user@host# show security zones
security-zone Untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/15.0;
    }
}
security-zone Trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/14.0;
    }
}
[edit]
user@host# show security address-book
book1 {
    address Sunnyvale 2001:db8:3::2/96;
    attach {
        zone Trust;
    }
}
    book2 {
        address Chicago 2001:db8:0::2/96;
        attach {
            zone Untrust;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ipv6-ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ipv6-ike-phase1-proposal dh-group group2
set security ike proposal ipv6-ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ipv6-ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ipv6-ike-phase1-policy mode aggressive
set security ike policy ipv6-ike-phase1-policy proposals ipv6-ike-phase1-proposal
set security ike policy ipv6-ike-phase1-policy pre-shared-key ascii-text 1111111111111111
set security ike gateway gw-chicago external-interface ge-0/0/15.0
set security ike gateway gw-chicago ike-policy ipv6-ike-phase1-policy
set security ike gateway gw-chicago address 2001:db8:0:1::1/96
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE:

1. Create the IKE Phase 1 proposal.

   ```
   [edit security ike]
   user@host# set proposal ipv6-ike-phase1-proposal
   ```

2. Define the IKE proposal authentication method.

   ```
   [edit security ike proposal ipv6-ike-phase1-proposal]
   user@host# set authentication-method pre-shared-keys
   ```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ipv6-ike-phase1-proposal]
user@host# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ipv6-ike-phase1-proposal]
user@host# set authentication-algorithm sha1
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ipv6-ike-phase1-proposal]
user@host# set encryption-algorithm aes-128-cbc
```

6. Create an IKE Phase 1 policy.

```
[edit security ike]
user@host# set policy ipv6-ike-phase1-policy
```

7. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ipv6-ike-phase1-policy]
user@host# set mode aggressive
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ipv6-ike-phase1-policy]
user@host# set proposals ipv6-ike-phase1-proposal
```

9. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ipv6-ike-phase1-policy]
user@host# set pre-shared-key ascii-text 1111111111111111
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@host# set gateway gw-chicago external-interface ge-0/0/15.0
```

11. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway gw-chicago]
user@host# set ike-policy ipv6-ike-phase1-policy
```

12. Assign an IP address to the IKE Phase 1 gateway.

```
[edit security ike gateway gw-chicago]
user@host# set address 2001:db8:1::1
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
proposal ipv6-ike-phase1-proposal {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
policy ipv6-ike-phase1-policy {
    mode ;
    proposals ipv6-ike-phase1-proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway gw-chicago {
    ike-policy ipv6-ike-phase1-policy;
    address 2001:db8:1::1;
    external-interface ge-0/0/15.0;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipv6-ipsec-phase2-proposal protocol esp
set security ipsec proposal ipv6-ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipv6-ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipv6-ipsec-phase2-policy proposals ipv6-ipsec-phase2-proposal
set security ipsec policy ipv6-ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn ipv6-ike-vpn-chicago ike gateway gw-chicago
set security ipsec vpn ipv6-ike-vpn-chicago ike ipv6-ipsec-policy ipsec-phase2-policy
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

   ```
   [edit]
   user@host# set security ipsec proposal ipv6-ipsec-phase2-proposal
   ```

2. Specify the IPsec Phase 2 proposal protocol.

   ```
   [edit security ipsec proposal ipv6- ipsec-phase2-proposal]
   user@host# set protocol esp
   ```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipv6-ipsec-phase2-proposal]
user@host# set authentication-algorithm hmac-sha1-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipv6-ipsec-phase2-proposal]
user@host# set encryption-algorithm aes-128-cbc
```

5. Create the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set policy ipv6-ipsec-phase2-policy
```

6. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipv6-ipsec-phase2-policy]
user@host# set proposals ipv6-ipsec-phase2-proposal
```

7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.

```
[edit security ipsec policy ipv6-ipsec-phase2-policy]
user@host# set perfect-forward-secrecy keys group2
```

8. Specify the IKE gateway.

```
[edit security ipsec]
user@host# set vpn ipv6-ike-vpn-chicago ike gateway gw-chicago
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn ipv6-ike-vpn-chicago ike ipsec-policy ipv6-ipsec-phase2-policy
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ipsec
proposal ipv6-ipsec-phase2-proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
}
policy ipv6-ipsec-phase2-policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipv6-ipsec-phase2-proposal;
}
vpn ipv6-ike-vpn-chicago {
    ike {
        gateway gw-chicago;
        ipsec-policy ipv6-ipsec-phase2-policy;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Security Policies

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr match source-
address Sunnyvale
set security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr match destination-
address Chicago
set security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr match application
```

```
any
set security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr then permit tunnel ipsec-vpn ipv6-ike-vpn-chicago
set security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr then permit tunnel pair-policy ipv6-vpn-untr-tr
set security policies from-zone Untrust to-zone Trust policy ipv6-vpn-untr-tr match source-address Chicago
set security policies from-zone Untrust to-zone Trust policy ipv6-vpn-untr-tr match destination-address Sunnyvale
set security policies from-zone Untrust to-zone Trust policy ipv6-vpn-untr-tr match application any
set security policies from-zone Untrust to-zone Trust policy ipv6-vpn-untr-tr then permit tunnel ipsec-vpn ipv6-ike-vpn-chicago
set security policies from-zone Untrust to-zone Trust policy ipv6-vpn-untr-tr then permit tunnel pair-policy ipv6-vpn-tr-untr
set security policies from-zone Trust to-zone Untrust policy permit-any match source-address any
set security policies from-zone Trust to-zone Untrust policy permit-any match destination-address any
set security policies from-zone Trust to-zone Untrust policy permit-any match application any
set security policies from-zone Trust to-zone Untrust policy permit-any then permit
insert security policies from-zone Trust to-zone Untrust policy ipv6-vpn-tr-untr before policy permit-any
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure security policies:

1. Create the security policy to permit traffic from the Trust zone to the Untrust zone.

```
[edit security policies from-zone Trust to-zone Untrust]
user@host# set policy ipv6-vpn-tr-untr match source-address Sunnyvale
user@host# set policy ipv6-vpn-tr-untr match destination-address Chicago
user@host# set policy ipv6-vpn-tr-untr match application any
user@host# set policy ipv6-vpn-tr-untr then permit tunnel ipsec-vpn ipv6-ike-vpn-chicago
user@host# set policy ipv6-vpn-tr-untr then permit tunnel pair-policy ipv6-vpn-untr-tr
```

2. Create the security policy to permit traffic from the Untrust zone to the Trust zone.

```
[edit security policies from-zone Untrust to-zone Trust]
user@host# set policy ipv6-vpn-untr-tr match source-address Sunnyvale
user@host# set policy ipv6-vpn-untr-tr match destination-address Chicago
user@host# set policy ipv6-vpn-untr-tr match application any
user@host# set policy ipv6-vpn-untr-tr then permit tunnel ipsec-vpn ipv6-ike-vpn-chicago
user@host# set policy ipv6-vpn-untr-tr then permit tunnel pair-policy ipv6-vpn-tr-untr
```

3. Create the security policy to permit traffic from the Trust zone to the Untrust zone.

```
[edit security policies from-zone Trust to-zone Untrust]
user@host# set policy permit-any match source-address any
user@host# set policy permit-any match destination-address any
user@host# set policy permit-any match application any
user@host# set policy permit-any then permit
```

4. Reorder the security policies so that the vpn-tr-untr security policy is placed above the permit-any security policy.

```
[edit security policies from-zone Trust to-zone Untrust]
user@host# insert policy ipv6-vpn-tr-untr before policy permit-any
```

Results

From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone Trust to-zone Untrust {
    policy ipv6-vpn-tr-untr {
        match {
            source-address Sunnyvale;
            destination-address Chicago;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-vpn ipv6-ike-vpn-chicago;
                    pair-policy ipv6-vpn-untr-tr;
                }
            }
        }
    }
    policy permit-any {
        match {
            source-address any;
            destination-address any;
            application any;

        }
        then {
            permit
        }
    }
}
from-zone Untrust to-zone Trust {
    policy ipv6-vpn-untr-tr {
        match {
            source-address Chicago;
            destination-address Sunnyvale;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-vpn ipv6-ike-vpn-chicago;
                    pair-policy ipv6-vpn-tr-untr;
                }
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring TCP-MSS

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter commit from configuration mode.

```
set security flow tcp-mss ipsec-vpn mss 1350
```

Step-by-Step Procedure

To configure TCP-MSS information:

1. Configure TCP-MSS information.

```
[edit]
user@host# set security flow tcp-mss ipsec-vpn mss 1350
```

Results

From configuration mode, confirm your configuration by entering the show security flow command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security flow
tcp-mss {
    ipsec-vpn {
        mss 1350;
    }
}
```

If you are done configuring the device, enter commit from configuration mode.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

Before starting the verification process, you need to send traffic from a host in Sunnyvale to a host in Chicago. For policy-based VPNs, a separate host must generate the traffic; traffic initiated from the SRX Series Firewall will not match the VPN policy. We recommend that the test traffic be from a separate device on one side of the VPN to a second device on the other side of the VPN. For example, initiate ping from 2001:db8:3::2/96 to 2001:db8:0::2/96.

From operational mode, enter the `show security ike security-associations` command. After obtaining an index number from the command, use the `show security ike security-associations index` *`index_number`* `detail` command.

```
user@host> show security ike security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  Mode
5       2001:db8:1::1        UP     e48efd6a444853cf  0d09c59aafb720be  Aggressive
```

```
user@host> show security ike security-associations index 5 detail
IKE peer 2001:db8:1::1, Index 5,
  Role: Initiator, State: UP
  Initiator cookie: e48efd6a444853cf, Responder cookie: 0d09c59aafb720be
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 2001:db8:2::1:500, Remote: 2001:db8:1::1:500
```

```
Lifetime: Expires in 19518 seconds
Peer ike-id: not valid
Xauth assigned IP: 0.0.0.0
Algorithms:
 Authentication        : sha1
 Encryption            : aes-128-cbc
 Pseudo random function: hmac-sha1
Traffic statistics:
 Input  bytes  :                 1568
 Output bytes  :                 2748
 Input  packets:                    6
 Output packets:                   23
Flags: Caller notification sent
IPSec security associations: 5 created, 0 deleted
Phase 2 negotiations in progress: 1

  Negotiation type: Quick mode, Role: Initiator, Message ID: 2900338624
  Local: 2001:db8:2::1:500, Remote: 2001:db8:1::1:500
  Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Flags: Caller notification sent, Waiting for done
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 security associations (SAs). If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index` *`index_number`* `detail` command to get more information about the SA.
- Remote Address—Verify that the remote IP address is correct.
- State
  - UP—The Phase 1 SA has been established.
  - DOWN—There was a problem establishing the Phase 1 SA.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations index 5 detail` command lists additional information about the security association with an index number of 5:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Initiator and responder role information

  Troubleshooting is best performed on the peer using the responder role.
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

Verifying the IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *`index_number`* `detail` command.

```
user@host> show security ipsec security-associations
  total configured sa: 2
   ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  2     ESP:aes-128/sha1 14caf1d9 3597/ unlim   -   root 500   2001:db8:1::1
  2     ESP:aes-128/sha1 9a4db486 3597/ unlim   -   root 500   2001:db8:1::1
```

```
user@host> show security ipsec security-associations index 2 detail
  Virtual-system: Root
  Local Gateway: 2001:db8:2::1, Remote Gateway: 2001:db8:1::1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    DF-bit: clear
    Direction: inbound, SPI: 14caf1d9, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3440 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2813 seconds
    Mode: tunnel, Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64

    Direction: outbound, SPI: 9a4db486, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3440 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2813 seconds
    Mode: tunnel, Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 2. Use this value with the `show security ipsec security-associations index` command to get more information about this particular SA.

- There is one IPsec SA pair using port 500, which indicates that no NAT-traversal is implemented. (NAT-traversal uses port 4500 or another random high-number port.)

- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3597/unlim value indicates that the Phase 2 lifetime expires in 3597 seconds, and that no lifesize has been specified, which indicates that the lifetime is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.

- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U (up) or D (down) is listed.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 2 detail` command lists the following information:

- The local and remote identities make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common reasons for a Phase 2 failure. For policy-based VPNs, the proxy ID is derived from the security policy. The local and remote addresses are derived from the address book entries, and the service is derived from the application configured for the policy. If Phase 2 fails because of a proxy ID mismatch, you can use the policy to confirm which address book entries are configured. Verify that the addresses match the information being sent. Check the service to ensure that the ports match the information being sent.

  For some third-party vendors, the proxy ID must be manually entered to match.

SEE ALSO

Internet Key Exchange | 10

RELATED DOCUMENTATION

IPsec VPN Configuration Overview | 190

7

CHAPTER

Route Based VPN

Route-Based IPsec VPNs

IN THIS SECTION

A route-based VPN is a configuration in which an IPsec VPN tunnel created between two end points is referenced by a route that determines which traffic is sent through the tunnel based on a destination IP address.

Understanding Route-Based IPsec VPNs

With route-based VPNs, you can configure dozens of security policies to regulate traffic flowing through a single VPN tunnel between two sites, and there is just one set of IKE and IPsec SAs at work. Unlike policy-based VPNs, for route-based VPNs, a policy refers to a destination address, not a VPN tunnel. When Junos OS looks up a route to find the interface to use to send traffic to the packet's destination address, it finds a route through a secure tunnel interface (st0.*x*). The tunnel interface is bound to a specific VPN tunnel, and the traffic is routed to the tunnel if the policy action is permit.

A secure tunnel (st0) interface supports only one IPv4 address and one IPv6 address at the same time. This applies to all route-based VPNs. The `disable` option is not supported on st0 interfaces.

NOTE: A secure tunnel interface (st0) from st0.16000 to st0.16385 is reserved for Multinode High Availability and for HA control link encryption in Chassis Cluster. These interfaces are not user configurable interfaces. You can only use interfaces from st0.0 to st0.15999.

Examples of where route-based VPNs can be used:

- There are overlapping subnets or IP addresses between the two LANs.
- A hub-and-spoke VPN topology is used in the network, and spoke-to-spoke traffic is required.
- Primary and backup VPNs are required.

- A dynamic routing protocol (for example, OSPF, RIP, or BGP) is running across the VPN.

  Configuring RIP demand circuits over point-to-multipoint VPN interfaces is not supported.

We recommend that you use route-based VPN when you want to configure VPN between multiple remote sites. Route-based VPN allows for routing between the spokes between multiple remote sites; it is easier to configure, monitor, and troubleshoot.

SEE ALSO

Class of Service User Guide (Security Devices)

IPsec Overview | 20

Example: Configuring a Hub-and-Spoke VPN | 209

Example: Configuring a Policy-Based VPN | 268

Example: Configuring a Route-Based VPN

IN THIS SECTION

This example shows how to configure a route-based IPsec VPN to allow data to be securely transferred between two sites.

Requirements

This example uses the following hardware:

- Any SRX Series Firewall
  - Updated and revalidated using vSRX Virtual Firewall on Junos OS Release 20.4R1.

NOTE: Are you interested in getting hands-on experience with the topics and operations covered in this guide? Visit the IPsec Route-Based VPN demonstration in Juniper Networks Virtual Labs and reserve your free sandbox today! You'll find the IPsec VPN Route-Based sandbox in the Security category.

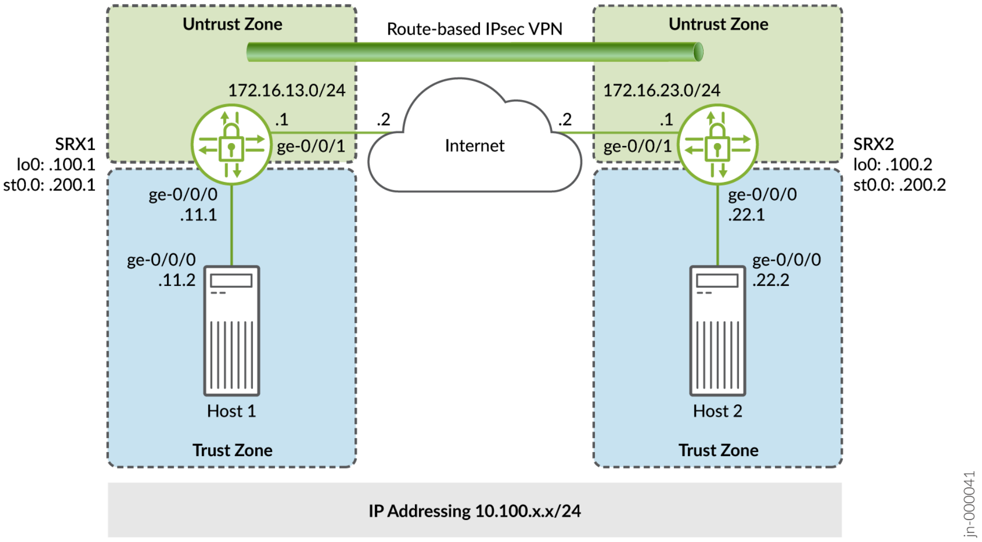
Before you begin, read "IPsec Overview" on page 20.

Overview

In this example, you configure a route-based VPN on SRX1 and SRX2. Host1 and Host2 use the VPN to send traffic securely over the Internet between both hosts.

Figure 32 on page 396 shows an example of a route-based VPN topology.

Figure 32: Route-Based VPN Topology



In this example, you configure interfaces, an IPv4 default route, and security zones. Then you configure IKE, IPsec, security policy, and TCP-MSS parameters. See Table 44 on page 397 through Table 48 on page 399 for specific configuration parameters used in this example.

Table 44: Interface, Static Route, Security Zone, and Security Policy Information for SRX1

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	10.100.11.1/24
	ge-0/0/1.0	172.16.13.1/24
	st0.0 (tunnel interface)	10.100.200.1/24
Static routes	10.100.22.0/24 0.0.0.0/0	The next hop is st0.0. The next hop is 172.16.13.2.
Security zones	trust	• The ge-0/0/0.0 interface is bound to this zone.
	untrust	• The ge-0/0/1.0 interface is bound to this zone.
	vpn	• The st0.0 interface is bound to this zone.

Table 45: IKE Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	standard	• Authentication method: pre-shared-keys
Policy	IKE-POL	• Mode: main • Proposal reference: standard • IKE policy authentication method: pre-shared-keys

Table 45: IKE Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Gateway	IKE-GW	• IKE policy reference: IKE-POL • External interface: ge-0/0/1 • Gateway address: 172.16.23.1

Table 46: IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	standard	• Using default configuration
Policy	IPSEC-POL	• Proposal reference: standard
VPN	VPN-to-Host2	• IKE gateway reference: IKE-GW • IPsec policy reference: IPSEC-POL • Bind to interface: st0.0 • establish-tunnels immediately

Table 47: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
The security policy permits traffic from the trust zone to the VPN zone.	VPN-OUT	• Match criteria: • source-address Host1-Net • destination-address Host2-Net • application any • Action: permit

Table 47: Security Policy Configuration Parameters *(Continued)*

Purpose	Name	Configuration Parameters
The security policy permits traffic from the VPN zone to the trust zone.	VPN-IN	• Match criteria: • source-address Host2-Net • destination-address Host1-Net • application any • Action: permit

Table 48: TCP-MSS Configuration Parameters

Purpose	Configuration Parameters
TCP-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the MTU limits on a network. For VPN traffic, the IPsec encapsulation overhead, along with the IP and the frame overhead, can cause the resulting ESP packet to exceed the MTU of the physical interface, which causes fragmentation. Fragmentation increases bandwidth and the device resources. We recommend a value of 1350 as the starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.	MSS value: 1350

Configuration

IN THIS SECTION

Configure Basic Network and Security Zone Information

CLI Quick Configuration

To quickly configure this section of the example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.100.11.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.13.1/24
set interfaces lo0 unit 0 family inet address 10.100.100.1/32
set interfaces st0 unit 0 family inet address 10.100.200.1/24
set routing-options static route 10.100.22.0/24 next-hop st0.0
set routing-options static route 0.0.0.0/0 next-hop 172.16.13.2
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone VPN host-inbound-traffic system-services ping
set security zones security-zone VPN interfaces st0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure interface, static route, and security zone information:

1. Configure the interfaces.

```
[edit]
user@SRX1# set interfaces ge-0/0/0 unit 0 family inet address 10.100.11.1/24
user@SRX1# set interfaces ge-0/0/1 unit 0 family inet address 172.16.13.1/24
user@SRX1# set interfaces lo0 unit 0 family inet address 10.100.100.1/32
user@SRX1# set interfaces st0 unit 0 family inet address 10.100.200.1/24
```

2. Configure the static routes.

```
[edit]
user@SRX1# set routing-options static route 10.100.22.0/24 next-hop st0.0
user@SRX1# set routing-options static route 0.0.0.0/0 next-hop 172.16.13.2
```

3. Assign the Internet facing interface to the untrust security zone.

```
[edit security zones security-zone untrust]
user@SRX1# set interfaces ge-0/0/1.0
```

4. Specify the allowed system services for the untrust security zone.

```
[edit security zones security-zone untrust]
user@SRX1# set host-inbound-traffic system-services ike
user@SRX1# set host-inbound-traffic system-services ping
```

5. Assign the Host1 facing interface to the trust security zone.

```
[edit security zones security-zone trust]
user@SRX1# set interfaces ge-0/0/0.0
```

6. Specify the allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@SRX1# set host-inbound-traffic system-services all
```

7. Assign the secure tunnel interface to the VPN security zone.

```
[edit security zones security-zone VPN]
user@SRX1# set interfaces st0.0
```

8. Specify the allowed system services for the VPN security zone.

```
[edit security zones security-zone VPN]
user@SRX1# set host-inbound-traffic system-services ping
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security zones` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@SRX1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.100.11.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 172.16.13.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.100.100.1/32;
        }
    }
}
```

```
st0 {
    unit 0 {
        family inet {
            address 10.100.200.1/24;
        }
    }
}
```

```
[edit]
user@SRX1# show routing-options
static {
    route 10.100.22.0/24 next-hop st0.0;
    route 0.0.0.0/0 next-hop 172.16.13.2;
}
```

```
[edit]
user@SRX1# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
security-zone VPN {
    host-inbound-traffic {
        system-services {
            ping;
        }
    }
    interfaces {
        st0.0;
    }
}
```

Configuring IKE

CLI Quick Configuration

To quickly configure this section of the example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```

set security ike proposal standard authentication-method pre-shared-keys
set security ike policy IKE-POL mode main
set security ike policy IKE-POL proposals standard
set security ike policy IKE-POL pre-shared-key ascii-text $ABC123
set security ike gateway IKE-GW ike-policy IKE-POL
set security ike gateway IKE-GW address 172.16.23.1
set security ike gateway IKE-GW external-interface ge-0/0/1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see CLI User Guide.

To configure IKE:

1. Create the IKE proposal.

   ```
   [edit security ike]
   user@SRX1# set proposal standard
   ```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal standard]
user@SRX1# set authentication-method pre-shared-keys
```

3. Create an IKE policy.

```
[edit security ike]
user@SRX1# set policy IKE-POL
```

4. Set the IKE policy mode.

```
[edit security ike policy IKE-POL]
user@SRX1# set mode main
```

5. Specify a reference to the IKE proposal.

```
[edit security ike policy IKE-POL]
user@SRX1# set proposals standard
```

6. Define the IKE policy authentication method.

```
[edit security ike policy IKE-POL]
user@SRX1# set pre-shared-key ascii-text $ABC123
```

7. Create an IKE gateway and define its external interface.

```
[edit security ike]
user@SRX1# set gateway IKE-GW external-interface ge-0/0/1
```

8. Define the IKE policy reference.

```
[edit security ike gateway IKE-GW]
user@SRX1# set ike-policy IKE-POL
```

9. Define the IKE gateway address.

   ```
   [edit security ike gateway IKE-GW]
   user@SRX1# set address 172.16.23.1
   ```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@SRX1# show security ike
proposal standard {
    authentication-method pre-shared-keys;
}
policy IKE-POL {
    mode main;
    proposals standard;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway IKE-GW {
    ike-policy IKE-POL;
    address 172.16.23.1;
    external-interface ge-0/0/1;
}
```

Configuring IPsec

CLI Quick Configuration

To quickly configure this section of the example for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```

set security ipsec proposal standard
set security ipsec policy IPSEC-POL proposals standard
```

```
set security ipsec vpn VPN-to-Host2 bind-interface st0.0
set security ipsec vpn VPN-to-Host2 ike gateway IKE-GW
set security ipsec vpn VPN-to-Host2 ike ipsec-policy IPSEC-POL
set security ipsec vpn VPN-to-Host2 establish-tunnels immediately
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure IPsec:

1. Create an IPsec proposal.

   ```
   [edit]
   user@SRX1# set security ipsec proposal standard
   ```

2. Create the IPsec policy.

   ```
   [edit security ipsec]
   user@SRX1# set policy IPSEC-POL
   ```

3. Specify the IPsec proposal reference.

   ```
   [edit security ipsec policy IPSEC-POL]
   user@SRX1# set proposals standard
   ```

4. Specify the IKE gateway.

   ```
   [edit security ipsec]
   user@SRX1# set vpn VPN-to-Host2 ike gateway IKE-GW
   ```

5. Specify the IPsec policy.

   ```
   [edit security ipsec]
   user@host# set vpn VPN-to-Host2 ike ipsec-policy IPSEC-POL
   ```

6. Specify the interface to bind.

```
[edit security ipsec]
user@host# set vpn VPN-to-Host2 bind-interface st0.0
```

7. Configure the tunnel to establish immediately.

```
[edit security ipsec]
user@host# set vpn VPN-to-Host2 establish-tunnels immediately
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ipsec
proposal standard;
policy IPSEC-POL {
    proposals standard;
}
vpn VPN-to-Host2 {
    bind-interface st0.0;
    ike {
        gateway IKE-GW;
        ipsec-policy IPSEC-POL;
    }
    establish-tunnels immediately;
}
```

Configuring Security Policies

CLI Quick Configuration

To quickly configure security policies for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set security address-book Host1 address Host1-Net 10.100.11.0/24
set security address-book Host1 attach zone trust
set security address-book Host2 address Host2-Net 10.100.22.0/24
set security address-book Host2 attach zone VPN
set security policies from-zone trust to-zone untrust policy default-permit match source-address any
set security policies from-zone trust to-zone untrust policy default-permit match destination-address any
set security policies from-zone trust to-zone untrust policy default-permit match application any
set security policies from-zone trust to-zone untrust policy default-permit then permit
set security policies from-zone trust to-zone VPN policy VPN-OUT match source-address Host1-Net
set security policies from-zone trust to-zone VPN policy VPN-OUT match destination-address Host2-Net
set security policies from-zone trust to-zone VPN policy VPN-OUT match application any
set security policies from-zone trust to-zone VPN policy VPN-OUT then permit
set security policies from-zone VPN to-zone trust policy VPN-IN match source-address Host2-Net
set security policies from-zone VPN to-zone trust policy VPN-IN match destination-address Host1-Net
set security policies from-zone VPN to-zone trust policy VPN-IN match application any
set security policies from-zone VPN to-zone trust policy VPN-IN then permit
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure security policies:

1. Create address book entries for the networks that will be used in the security policies.

   ```
   [edit]
   user@SRX1# set security address-book Host1 address Host1-Net 10.100.11.0/24
   user@SRX1# set security address-book Host1 attach zone trust
   user@SRX1# set security address-book Host2 address Host2-Net 10.100.22.0/24
   user@SRX1# set security address-book Host2 attach zone VPN
   ```

2. Create a security policy to permit traffic from the trust zone to the untrust zone for traffic to the Internet.

   ```
   [edit security policies from-zone trust to-zone untrust]
   user@SRX1# set policy default-permit match source-address any
   user@SRX1# set policy default-permit match destination-address any
   user@SRX1# set policy default-permit match application any
   user@SRX1# set policy default-permit then permit
   ```

3. Create a security policy to permit traffic from Host1 in the trust zone destined to Host2 in the VPN zone.

   ```
   [edit security policies from-zone trust to-zone VPN]
   user@SRX1# set policy VPN-OUT match source-address Host1-Net
   user@SRX1# set policy VPN-OUT match destination-address Host2-Net
   user@SRX1# set policy VPN-OUT match application any
   user@SRX1# set policy VPN-OUT then permit
   ```

4. Create a security policy to permit traffic from Host2 in the VPN zone to Host1 in the trust zone.

   ```
   [edit security policies from-zone VPN to-zone trust]
   user@host# set policy VPN-IN match source-address Host2-Net
   user@host# set policy VPN-IN match destination-address Host1-Net
   user@host# set policy VPN-IN match application any
   user@host# set policy VPN-IN then permit
   ```

Results

From configuration mode, confirm your configuration by entering the `show security address-book` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security address-book
Host1 {
    address Host1-Net 10.100.11.0/24;
    attach {
        zone trust;
```

```
    }
}
Host2 {
    address Host2-Net 10.100.22.0/24;
    attach {
        zone VPN;
    }
}
user@host# show security policies
from-zone trust to-zone untrust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone trust to-zone VPN {
    policy VPN-OUT {
        match {
            source-address Host1-Net;
            destination-address Host2-Net;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone VPN to-zone trust {
    policy VPN-IN {
        match {
            source-address Host2-Net;
            destination-address Host1-Net;
            application any;
        }
        then {
            permit;
        }
    }
}
```

Configuring TCP-MSS

CLI Quick Configuration

To quickly configure the TCP MSS for SRX1, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set security flow tcp-mss ipsec-vpn mss 1350
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the CLI User Guide.

To configure TCP-MSS information:

1. Configure the TCP-MSS information.

   ```
   [edit]
   user@SRX1# set security flow tcp-mss ipsec-vpn mss 1350
   ```

Results

From configuration mode, confirm your configuration by entering the `show security flow` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@SRX1# show security flow
tcp-mss {
    ipsec-vpn {
        mss 1350;
```

```
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring SRX2

CLI Quick Configuration

For reference, the configuration for the SRX2 is provided.

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```

set security ike proposal standard authentication-method pre-shared-keys
set security ike policy IKE-POL mode main
set security ike policy IKE-POL proposals standard
set security ike policy IKE-POL pre-shared-key ascii-text $ABC123
set security ike gateway IKE-GW ike-policy IKE-POL
set security ike gateway IKE-GW address 172.16.13.1
set security ike gateway IKE-GW external-interface ge-0/0/1
set security ipsec proposal standard
set security ipsec policy IPSEC-POL proposals standard
set security ipsec vpn VPN-to-Host1 bind-interface st0.0
set security ipsec vpn VPN-to-Host1 ike gateway IKE-GW
set security ipsec vpn VPN-to-Host1 ike ipsec-policy IPSEC-POL
set security ipsec vpn VPN-to-Host1 establish-tunnels immediately
set security address-book Host1 address Host1-Net 10.100.11.0/24
set security address-book Host1 attach zone VPN
set security address-book Host2 address Host2-Net 10.100.22.0/24
set security address-book Host2 attach zone trust
set security flow tcp-mss ipsec-vpn mss 1350
set security policies from-zone trust to-zone untrust policy default-permit match source-address
any
set security policies from-zone trust to-zone untrust policy default-permit match destination-
address any
set security policies from-zone trust to-zone untrust policy default-permit match application any
set security policies from-zone trust to-zone untrust policy default-permit then permit
set security policies from-zone trust to-zone VPN policy VPN-OUT match source-address Host2-Net
set security policies from-zone trust to-zone VPN policy VPN-OUT match destination-address Host1-
Net
set security policies from-zone trust to-zone VPN policy VPN-OUT match application any
set security policies from-zone trust to-zone VPN policy VPN-OUT then permit
set security policies from-zone VPN to-zone trust policy VPN-IN match source-address Host1-Net
set security policies from-zone VPN to-zone trust policy VPN-IN match destination-address Host2-
Net
set security policies from-zone VPN to-zone trust policy VPN-IN match application any
set security policies from-zone VPN to-zone trust policy VPN-IN then permit
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone VPN host-inbound-traffic system-services ping
set security zones security-zone VPN interfaces st0.0
set interfaces ge-0/0/0 unit 0 family inet address 10.100.22.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.23.1/24
set interfaces lo0 unit 0 family inet address 10.100.100.2/32
set interfaces st0 unit 0 family inet address 10.100.200.2/24
set routing-options static route 10.100.11.0/24 next-hop st0.0
set routing-options static route 0.0.0.0/0 next-hop 172.16.23.2
```

Verification

IN THIS SECTION

Perform these tasks to confirm that the configuration is working properly:

Verify the IKE Status

Purpose

Verify the IKE status.

Action

From operational mode, enter the `show security ike security-associations` command. After obtaining an index number from the command, use the `show security ike security-associations index` *`index_number`* `detail` command.

```
user@SRX1> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
1859340 UP     b153dc24ec214da9  5af2ee0c2043041a  Main           172.16.23.1
```

```
user@SRX1> show security ike security-associations index 1859340 detail
IKE peer 172.16.23.1, Index 1859340, Gateway Name: IKE-GW
  Role: Responder, State: UP
  Initiator cookie: b153dc24ec214da9, Responder cookie: 5af2ee0c2043041a
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 172.16.13.1:500, Remote: 172.16.23.1:500
  Lifetime: Expires in 23038 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
  Remote Access Client Info: Unknown Client
  Peer ike-id: 172.16.23.1
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :                 1236
   Output bytes  :                  868
   Input  packets:                    9
   Output packets:                    5
   Input  fragmentated packets:       0
   Output fragmentated packets:       0
  IPSec security associations: 2 created, 2 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 172.16.13.1:500, Remote: 172.16.23.1:500
    Local identity: 172.16.13.1
    Remote identity: 172.16.23.1
    Flags: IKE SA is created
```

Meaning

The `show security ike security-associations` command lists all active IKE SAs. If no SAs are listed, there was a problem with IKE establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.

- Remote Address—Verify that the remote IP address is correct.

- State

  - UP—The IKE SA has been established.

  - DOWN—There was a problem establishing the IKE SA.

- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)

- IKE policy parameters

- Preshared key information

- Proposal parameters (must match on both peers)

The `show security ike security-associations index 1859340 detail` command lists additional information about the security association with an index number of 1859340:

- Authentication and encryption algorithms used

- lifetime

- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information

  Troubleshooting is best performed on the peer using the responder role.
- Initiator and responder information
- Number of IPsec SAs created
- Number of negotiations in progress

Verify the IPsec Status

Purpose

Verify the IPsec status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *index_number* `detail` command.

```
user@SRX1> show security ipsec security-associations
  Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131074 ESP:3des/sha1   912f9063 3403/ unlim  -   root 500   172.16.23.1
  >131074 ESP:3des/sha1   71dbaa56 3403/ unlim  -   root 500   172.16.23.1
```

```
user@SRX1> show security ipsec security-associations index 131074 detail
ID: 131074 Virtual-system: root, VPN Name: VPN-to-Host2
  Local Gateway: 172.16.13.1, Remote Gateway: 172.16.23.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0
  Port: 500, Nego#: 26, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Fri Jul 23 2021 10:46:34 -0700: IPSec SA negotiation successfully completed (23 times)
    Fri Jul 23 2021 09:07:24 -0700: IKE SA negotiation successfully completed (3 times)
    Thu Jul 22 2021 16:34:17 -0700: Negotiation failed with INVALID_SYNTAX error (3 times)
    Thu Jul 22 2021 16:33:50 -0700: Tunnel configuration changed. Corresponding IKE/IPSec SAs
are deleted (1 times)
    Thu Jul 22 2021 16:23:49 -0700: IPSec SA negotiation successfully completed (2 times)
    Thu Jul 22 2021 15:34:12
    : IPSec SA delete payload received from peer, corresponding IPSec SAs cleared (1 times)
    Thu Jul 22 2021 15:33:25 -0700: IPSec SA negotiation successfully completed (1 times)
    Thu Jul 22 2021 15:33:25
    : Tunnel is ready. Waiting for trigger event or peer to trigger negotiation (1 times)
    Thu Jul 22 2021 15:33:25 -0700: External interface's address received. Information updated
(1 times)
    Thu Jul 22 2021 15:33:25 -0700: Bind-interface's zone received. Information updated (1 times)
    Thu Jul 22 2021 10:34:55 -0700: IKE SA negotiation successfully completed (1 times)
    Thu Jul 22 2021 10:34:46 -0700: No response from peer. Negotiation failed (16 times)
  Direction: inbound, SPI: 912f9063, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3302 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2729 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 71dbaa56, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3302 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2729 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 131074. Use this value with the `show security ipsec security-associations index` command to get more information about this particular SA.

- There is one IPsec SA pair using port 500, which indicates that no NAT-traversal is implemented. (NAT-traversal uses port 4500 or another random high-number port.)

- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3403/ unlim value indicates that the lifetime expires in 3403 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Lifetime can differ from lifetime, as IPsec is not dependent on IKE after the VPN is up.

- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 131074 detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a IPsec failure. If no IPsec SA is listed, confirm that IPsec proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.
- Another common reason for IPsec failure is not specifying the ST interface binding. If IPsec cannot complete, check the kmd log or set trace options.

Test Traffic Flow Across the VPN

Purpose

Verify the traffic flow across the VPN.

Action

Use the `ping` command from the Host1 device to test traffic flow to Host2.

```
user@Host1> ping 10.100.22.1 rapid count 100
PING 10.100.22.1 (10.100.22.1): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!
--- 10.100.22.1 ping statistics ---
100 packets transmitted, 100 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.146/3.824/6.193/0.402 ms
```

Meaning

If the ping command fails from Host1, there might be a problem with the routing, security policies, end host, or encryption and decryption of ESP packets.

Review Statistics and Errors for an IPsec Security Association

Purpose

Review ESP and authentication header counters and errors for an IPsec security association.

Action

From operational mode, enter the `show security ipsec statistics index` *`index_number`* command, using the index number of the VPN for which you want to see statistics.

```
user@SRX1> show security ipsec statistics index 131074
ESP Statistics:
  Encrypted bytes:           13600
  Decrypted bytes:            8400
  Encrypted packets:           100
  Decrypted packets:           100
AH Statistics:
  Input bytes:                   0
  Output bytes:                  0
  Input packets:                 0
  Output packets:                0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

You can also use the `show security ipsec statistics` command to review statistics and errors for all SAs.

To clear all IPsec statistics, use the `clear security ipsec statistics` command.

Meaning

If you see packet loss issues across a VPN, run the `show security ipsec statistics` or `show security ipsec statistics detail` command several times to confirm if the encrypted and decrypted packet counters are incrementing. Look in the command output for any incrementing error counters.

SEE ALSO

IPsec Overview | 20

Example: Configuring a Policy-Based VPN | 268

IPsec Route Based VPN in Juniper Networks Virtual Labs

Route-Based VPN with IKEv2

IN THIS SECTION

Internet Key Exchange version 2 (IKEv2) is an IPsec based tunneling protocol that provides a secure VPN communication channel between peer VPN devices and defines negotiation and authentication for IPsec security associations (SAs) in a protected manner.

Table 49 on page 421 describes the IPsec Radius xAuth or CP values.

Table 49: IPsec Radius xAuth or CP values

Radius Attribute	Attribute ID	Attribute Name	Vendor ID (Dictionary)	Vendor Attribute ID	Attribute Value	Type
Standard	8	Framed IP address	NA	NA	IP address	IPv4 address
Standard	88	Framed pool	NA	NA	Name	Text
Standard	100	Framed IPv6 pool	NA	NA	Name	Text
Vendor	26	Primary DNS	4874 (Juniper ERX)	4	IP address	IPv4 address

Table 49: IPsec Radius xAuth or CP values *(Continued)*

Radius Attribute	Attribute ID	Attribute Name	Vendor ID (Dictionary)	Vendor Attribute ID	Attribute Value	Type
Vendor	26	Secondary DNS	4874 (Juniper ERX)	5	IP address	IPv4 address
Vendor	26	Primary WINS (NBNS)	4874 (Juniper ERX)	6	IP address	IPv4 address
Vendor	26	Secondary WINS (NBNS)	4874 (Juniper ERX)	7	IP address	IPv4 address
Vendor	26	IPv6 primary DNS	4874 (Juniper ERX)	47	IP address	hex-string or octets
Vendor	26	IPv6 secondary DNS	4874 (Juniper ERX)	48	IP address	hex-string or octets

Example: Configuring a Route-Based VPN for IKEv2

IN THIS SECTION

This example shows how to configure a route-based IPsec VPN to allow data to be securely transferred between a branch office and a corporate office.

Requirements

This example uses the following hardware:

- SRX240 device
- SSG140 device

Before you begin, read "IPsec Overview" on page 20.

Overview

In this example, you configure a route-based VPN for a branch office in Chicago, Illinois, because you want to conserve tunnel resources but still get granular restrictions on VPN traffic. Users in the Chicago office will use the VPN to connect to their corporate headquarters in Sunnyvale, California.

In this example, you configure interfaces, an IPv4 default route, security zones, and address books. Then you configure IKE Phase 1, IPsec Phase 2, a security policy, and TCP-MSS parameters. See Table 50 on page 423 through Table 54 on page 426 for specific configuration parameters used in this example.

Table 50: Interface, Static Route, Security Zone, and Address Book Information

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	192.168.10.1/24
	ge-0/0/3.0	10.1.1.2/30
	st0.0 (tunnel interface)	10.11.11.10/24
Static routes	0.0.0.0/0 (default route)	The next hop is 10.1.1.1.
	192.168.168.0/24	The next hop is st0.0.
Security zones	trust	• All system services are allowed. • The ge-0/0/0.0 interface is bound to this zone.

Table 50: Interface, Static Route, Security Zone, and Address Book Information *(Continued)*

Feature	Name	Configuration Parameters
	untrust	- IKE is the only allowed system service. - The ge-0/0/3.0 interface is bound to this zone.
	vpn-chicago	The st0.0 interface is bound to this zone.
Address book entries	sunnyvale	- This address is for the trust zone's address book. - The address for this address book entry is 192.168.10.0/24.
	chicago	- This address is for the untrust zone's address book. - The address for this address book entry is 192.168.168.0/24.

Table 51: IKE Phase 1 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ike-phase1-proposal	- Authentication method: pre-shared-keys - Diffie-Hellman group: group2 - Authentication algorithm: sha1 - Encryption algorithm: aes-128-cbc

Table 51: IKE Phase 1 Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Policy	ike-phase1-policy	- Mode: main - Proposal reference: ike-phase1-proposal - IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw-chicago	- IKE policy reference: ike-phase1-policy - External interface: ge-0/0/3.0 - Gateway address: 10.2.2.2

Table 52: IPsec Phase 2 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipsec-phase2-proposal	- Protocol: esp - Authentication algorithm: hmac-sha1-96 - Encryption algorithm: aes-128-cbc
Policy	ipsec-phase2-policy	- Proposal reference: ipsec-phase2-proposal - PFS: Diffie-Hellman group2
VPN	ipsec-vpn-chicago	- IKE gateway reference: gw-chicago - IPsec policy reference: ipsec-phase2-policy - Bind to interface: st0.0

Table 53: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
The security policy permits traffic from the trust zone to the vpn-chicago zone.	vpn-tr-chi	• Match criteria: • source-address sunnyvale • destination-address chicago • application any • Action: permit
The security policy permits traffic from the vpn-chicago zone to the trust zone.	vpn-chi-tr	• Match criteria: • source-address chicago • destination-address sunnyvale • application any • Action: permit

Table 54: TCP-MSS Configuration Parameters

Purpose	Configuration Parameters
TCP-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the MTU limits on a network. For VPN traffic, the IPsec encapsulation overhead, along with the IP and frame overhead, can cause the resulting ESP packet to exceed the MTU of the physical interface, which causes fragmentation. Fragmentation increases bandwidth and device resources. We recommend a value of 1350 as the starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.	MSS value: 1350

Configuration

IN THIS SECTION

Configuring Interface, Static Route, Security Zone, and Address Book Information

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 192.168.10.1/24
set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
set interfaces st0 unit 0 family inet address 10.11.11.10/24
set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
set routing-options static route 192.168.168.0/24 next-hop st0.0
set security zones security-zone untrust interfaces ge-0/0/3.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust address-book address sunnyvale 192.168.10.0/24
set security zones security-zone vpn-chicago interfaces st0.0
set security zones security-zone vpn-chicago address-book address chicago 192.168.168.0/24
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure interface, static route, security zone, and address book information:

1. Configure Ethernet interface information.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 192.168.10.1/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
user@host# set interfaces st0 unit 0 family inet address 10.11.11.10/24
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 10.1.1.1
user@host# set routing-options static route 192.168.168.0/24 next-hop st0.0
```

3. Configure the untrust security zone.

```
[edit ]
user@host# edit security zones security-zone untrust
```

4. Assign an interface to the security zone.

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/3.0
```

5. Specify allowed system services for the security zone.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services ike
```

6. Configure the trust security zone.

```
[edit]
user@host# edit security zones security-zone trust
```

7. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/0.0
```

8. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```

9. Configure the address book entry for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set address-book address sunnyvale 192.168.10.0/24
```

10. Configure the vpn-chicago security zone.

```
[edit]
user@host# edit security zones security-zone vpn-chicago
```

11. Assign an interface to the security zone.

```
[edit security zones security-zone vpn-chicago]
user@host# set interfaces st0.0
```

12. Configure the address book entry for the vpn-chicago zone.

```
[edit security zones security-zone vpn-chicago]
user@host# set address-book address chicago 192.168.168.0/24
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security zones` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 192.168.10.1/24;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 10.1.1.2/30
        }
    }
}
st0{
    unit 0 {
        family inet {
            address 10.11.11.10/24
        }
    }
}
```

```
[edit]
user@host# show routing-options
static {
    route 0.0.0.0/0 next-hop 10.1.1.1;
    route 192.168.168.0/24 next-hop st0.0;
}
```

```
[edit]
user@host# show security zones
```

```
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone trust {
    address-book {
        address sunnyvale 192.168.10.0/24;
    }
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone vpn-chicago {
    host-inbound-traffic {
        address-book {
            address chicago 192.168.168.0/24;
        }
    }
    interfaces {
        st0.0;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposal dh-group group2
set security ike proposal ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-phase1-policy proposals ike-phase1-proposal
set security ike policy ike-phase1-policy pre-shared-key ascii-text “$ABC123”
set security ike gateway gw-chicago external-interface ge-0/0/3.0
set security ike gateway gw-chicago ike-policy ike-phase1-policy
set security ike gateway gw-chicago address 10.2.2.2
set security ike gateway gw-chicago version v2-only
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE:

1. Create the IKE Phase 1 proposal.

   ```
   [edit security ike]
   user@host# set proposal ike-phase1-proposal
   ```

2. Define the IKE proposal authentication method.

   ```
   [edit security ike proposal ike-phase1-proposal]
   user@host# set authentication-method pre-shared-keys
   ```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set authentication-algorithm sha1
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set encryption-algorithm aes-128-cbc
```

6. Create an IKE Phase 1 policy.

```
[edit security ike]
user@host# set policy ike-phase1-policy
```

7. Specify a reference to the IKE proposal.

```
[edit security ike policy ike-phase1-policy]
user@host# set proposals ike-phase1-proposal
```

8. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike-phase1-policy]
user@host# set pre-shared-key ascii-text “$ABC123”
```

9. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@host# set gateway gw-chicago external-interface ge-0/0/3.0
```

10. Define the IKE Phase 1 policy reference.

    ```
    [edit security ike gateway gw-chicago]
    user@host# set ike-policy ike-phase1-policy
    ```

11. Define the IKE Phase 1 gateway address.

    ```
    [edit security ike gateway gw-chicago]
    user@host# set address 10.2.2.2
    ```

12. Define the IKE Phase 1 gateway version.

    ```
    [edit security ike gateway gw-chicago]
    user@host# set version v2-only
    ```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
proposal ike-phase1-proposal {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
policy ike-phase1-policy {
    proposals ike-phase1-proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway gw-chicago {
    ike-policy ike-phase1-policy;
    address 10.2.2.2;
    external-interface ge-0/0/3.0;
```

```
    version v2-only;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipsec-phase2-proposal protocol esp
set security ipsec proposal ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipsec-phase2-policy proposals ipsec-phase2-proposal
set security ipsec policy ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn ipsec-vpn-chicago ike gateway gw-chicago
set security ipsec vpn ipsec-vpn-chicago ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn ipsec-vpn-chicago bind-interface st0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

   ```
   [edit]
   user@host# set security ipsec proposal ipsec-phase2-proposal
   ```

2. Specify the IPsec Phase 2 proposal protocol.

   ```
   [edit security ipsec proposal ipsec-phase2-proposal]
   user@host# set protocol esp
   ```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@host# set authentication-algorithm hmac-sha1-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@host# set encryption-algorithm aes-128-cbc
```

5. Create the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set policy ipsec-phase2-policy
```

6. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec-phase2-policy]
user@host# set proposals ipsec-phase2-proposal
```

7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.

```
[edit security ipsec policy ipsec-phase2-policy]
user@host# set perfect-forward-secrecy keys group2
```

8. Specify the IKE gateway.

```
[edit security ipsec]
user@host# set vpn ipsec-vpn-chicago ike gateway gw-chicago
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn ipsec-vpn-chicago ike ipsec-policy ipsec-phase2-policy
```

10. Specify the interface to bind.

```
[edit security ipsec]
user@host# set vpn ipsec-vpn-chicago bind-interface st0.0
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ipsec
proposal ipsec-phase2-proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
}
policy ipsec-phase2-policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec-phase2-proposal;
}
vpn ipsec-vpn-chicago {
    bind-interface st0.0;
    ike {
        gateway gw-chicago;
        ipsec-policy ipsec-phase2-policy;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Security Policies

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match source-address sunnyvale
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match destination-address chicago
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match application any
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi then permit
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match source-address chicago
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match destination-address sunnyvale
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match application any
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr then permit
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure security policies:

1. Create the security policy to permit traffic from the trust zone to the vpn-chicago zone.

   ```
   [edit security policies from-zone trust to-zone vpn-chicago]
   user@host# set policy vpn-tr-chi match source-address sunnyvale
   user@host# set policy vpn-tr-chi match destination-address chicago
   user@host# set policy vpn-tr-chi match application any
   user@host# set policy vpn-tr-chi then permit
   ```

2. Create the security policy to permit traffic from the vpn-chicago zone to the trust zone.

```
[edit security policies from-zone vpn-chicago to-zone trust]
user@host# set policy vpn-chi-tr match source-address sunnyvale
user@host# set policy vpn-chi-tr match destination-address chicago
user@host# set policy vpn-chi-tr match application any
user@host# set policy vpn-chi-tr then permit
```

Results

From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone trust to-zone vpn-chicago {
    policy vpn-tr-vpn {
        match {
            source-address sunnyvale;
            destination-address chicago;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn-chicago to-zone trust {
    policy vpn-tr-vpn {
        match {
            source-address chicago;
            destination-address sunnyvale;
            application any;
        }
        then {
            permit;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring TCP-MSS

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security flow tcp-mss ipsec-vpn mss 1350
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure TCP-MSS information:

1. Configure TCP-MSS information.

   ```
   [edit]
   user@host# set security flow tcp-mss ipsec-vpn mss 1350
   ```

Results

From configuration mode, confirm your configuration by entering the `show security flow` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security flow
tcp-mss {
    ipsec-vpn {
        mss 1350;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the SSG Series Device

CLI Quick Configuration

For reference, the configuration for the SSG Series device is provided. For information about configuring SSG Series devices, see the *Concepts & Examples ScreenOS Reference Guide*, which is located at https://www.juniper.net/documentation.

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set zone name vpn-chicago
set interface ethernet0/6 zone Trust
set interface ethernet0/0 zone Untrust
set interface tunnel.1 zone vpn-chicago
set interface ethernet0/6 ip 192.168.168.1/24
set interface ethernet0/6 route
set interface ethernet0/0 ip 10.2.2.2/30
set interface ethernet0/0 route
set interface tunnel.1 ip 10.11.11.11/24
set flow tcp-mss 1350
set address Trust “192.168.168-net” 192.168.168.0 255.255.255.0
set address vpn-chicago "192.168.10-net" 192.168.10.0 255.255.255.0
set ike gateway corp-ike address 10.1.1.2 IKEv2 outgoing-interface ethernet0/0 preshare
395psksecr3t sec-level standard
set vpn corp-vpn gateway corp-ike replay tunnel idletime 0 sec-level standard
set vpn corp-vpn monitor optimized rekey
set vpn corp-vpn bind interface tunnel.1
set policy from Trust to Untrust “ANY” “ANY” “ANY” nat src permit
set policy from Trust to vpn-chicago “192.168.168-net” “192.168.10-net” “ANY” permit
set policy from vpn-chicago to Trust “192.168.10-net” “192.168.168-net” “ANY” permit
set route 192.168.10.0/24 interface tunnel.1
set route 0.0.0.0/0 interface ethernet0/0 gateway 10.2.2.1
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

Before starting the verification process, you need to send traffic from a host in the 192.168.10/24 network to a host in the 192.168.168/24 network. For route-based VPNs, traffic can be initiated by the SRX Series Firewall through the tunnel. We recommend that when testing IPsec tunnels, test traffic be sent from a separate device on one side of the VPN to a second device on the other side of the VPN. For example, initiate a ping from 192.168.10.10 to 192.168.168.10.

From operational mode, enter the `show security ike security-associations` command. After obtaining an index number from the command, use the `show security ike security-associations index` *`index_number`* `detail` command.

```
user@host> show security ike security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  Mode
1       10.2.2.2         UP     744a594d957dd513  1e1307db82f58387  IKEv2
```

```
user@host> show security ike security-associations index 1 detail
IKE peer 10.2.2.2, Index 1,
  Role: Responder, State: UP
  Initiator cookie: 744a594d957dd513, Responder cookie: 1e1307db82f58387
```

```
Exchange type: IKEv2, Authentication method: Pre-shared-keys
Local: 10.1.1.2:500, Remote: 10.2.2.2:500
Lifetime: Expires in 28570 seconds
Algorithms:
 Authentication        : sha1
 Encryption            : aes-cbc (128 bits)
 Pseudo random function: hmac-sha1
Traffic statistics:
 Input bytes    :                 852
 Output bytes   :                 940
 Input packets  :                   5
 Output packets :                   5
Flags: Caller notification sent
IPSec security associations: 1 created, 0 deleted
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.
- Remote Address—Verify that the remote IP address is correct.
- State
  - UP—The Phase 1 SA has been established.
  - DOWN—There was a problem establishing the Phase 1 SA.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets).
- IKE policy parameters.
- Preshared key information.
- Phase 1 proposal parameters (must match on both peers).

The `show security ike security-associations index 1 detail` command lists additional information about the SA with an index number of 1:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information

  Troubleshooting is best performed on the peer using the responder role.
- Initiator and responder information
- Number of IPsec SAs created

Verifying the IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *`index_number`* `detail` command.

```
user@host> show security ipsec security-associations
  total configured sa: 2
  ID    Gateway     Port  Algorithm       SPI      Life:sec/kb  Mon vsys
  <16384 10.2.2.2    500   ESP:aes-128/sha1   76d64d1d 3363/ unlim   -   0
  >16384 10.2.2.2    500   ESP:aes-128/sha1   a1024ee2 3363/ unlim   -   0
```

```
user@host> show security ipsec security-associations index 16384 detail
  Virtual-system: Root
  Local Gateway: 10.1.1.2, Remote Gateway: 10.2.2.2
  Local Identity: ipv4_subnet(any:0,[0..7]=192.168.10.0/24)
  Remote Identity: ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
  Version: IKEv2

    DF-bit: clear
```

```
   Direction: inbound, SPI: 1993755933, AUX-SPI: 0
   Hard lifetime: Expires in 3352 seconds
   Lifesize Remaining: Unlimited
   Soft lifetime: Expires in 2775 seconds
   Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
   Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
   Anti-replay service: enabled, Replay window size: 32

   Direction: outbound, SPI: 2701283042, AUX-SPI: 0
   Hard lifetime: Expires in 3352 seconds
   Lifesize Remaining: Unlimited
   Soft lifetime: Expires in 2775 seconds
   Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
   Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc
(128 bits)
   Anti-replay service: enabled, Replay window size: 32
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 16384. Use this value with the `show security ipsec security-associations index` command to get more information about this particular SA.
- There is one IPsec SA pair using port 500.
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3363/ unlim value indicates that the Phase 2 lifetime expires in 3363 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, because Phase 2 is not dependent on Phase 1 after the VPN is up.
- The vsys is the root system, and it is always listed as 0.
- The IKEv2 allows connections from a version 2 peer and will initiate a version 2 negotiation.

The output from the `show security ipsec security-associations index 16384 detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each

IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.

- Another common reason for Phase 2 failure is not specifying the ST interface binding. If IPsec cannot complete, check the kmd log or set trace options.

Reviewing Statistics and Errors for an IPsec Security Association

Purpose

Review ESP and authentication header counters and errors for an IPsec SA.

Action

From operational mode, enter the `show security ipsec statistics index` *`index_number`* command, using the index number of the VPN for which you want to see statistics.

```
user@host> show security ipsec statistics index 16384
ESP Statistics:
  Encrypted bytes:             920
  Decrypted bytes:            6208
  Encrypted packets:             5
  Decrypted packets:            87
AH Statistics:
  Input bytes:                   0
  Output bytes:                  0
  Input packets:                 0
  Output packets:                0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

You can also use the `show security ipsec statistics` command to review statistics and errors for all SAs.

To clear all IPsec statistics, use the `clear security ipsec statistics` command.

Meaning

If you see packet loss issues across a VPN, you can run the `show security ipsec statistics` or `show security ipsec statistics detail` command several times to confirm that the encrypted and decrypted packet counters are incrementing. You should also check that the other error counters are incrementing.

Testing Traffic Flow Across the VPN

Purpose

Verify the traffic flow across the VPN.

Action

You can use the `ping` command from the SRX Series Firewall to test traffic flow to a remote host PC. Make sure that you specify the source interface so that the route lookup is correct and the appropriate security zones are referenced during policy lookup.

From operational mode, enter the `ping` command.

```
ssg-> ping 192.168.168.10 interface ge-0/0/0 count 5
PING 192.168.168.10 (192.168.168.10): 56 data bytes
64 bytes from 192.168.168.10: icmp_seq=0 ttl=127 time=8.287 ms
64 bytes from 192.168.168.10: icmp_seq=1 ttl=127 time=4.119 ms
64 bytes from 192.168.168.10: icmp_seq=2 ttl=127 time=5.399 ms
64 bytes from 192.168.168.10: icmp_seq=3 ttl=127 time=4.361 ms
64 bytes from 192.168.168.10: icmp_seq=4 ttl=127 time=5.137 ms

--- 192.168.168.10 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.119/5.461/8.287/1.490 ms
```

You can also use the `ping` command from the SSG Series device.

```
user@host> ping 192.168.10.10 from ethernet0/6
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 192.168.10.10, timeout is 1 seconds from ethernet0/6
!!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=4/4/5 ms
```

Meaning

If the `ping` command fails from the SRX Series or SSG Series device, there might be a problem with the routing, security policies, end host, or encryption and decryption of ESP packets.

SEE ALSO

IPsec Overview | 20
Example: Configuring a Hub-and-Spoke VPN | 209
Example: Configuring a Policy-Based VPN

Example: Configuring the SRX Series for Pico Cell Provisioning with IKEv2 Configuration Payload

IN THIS SECTION

In networks where many devices are being deployed, managing the network needs to be simple. The IKEv2 configuration payload feature supports the provisioning of these devices without touching either the device configuration or the SRX Series configuration. This example shows how to configure an SRX Series to support pico cell provisioning using the IKEv2 configuration payload feature.

Requirements

This example uses the following hardware and software components:

- Two SRX Series Firewalls configured in a chassis cluster
- One SRX Series Firewall configured as an intermediate router
- Two pico cell clients
- One RADIUS server configured with pico cell client provisioning information
- Junos OS Release 12.1X46-D10 or later for IKEv2 configuration payload support

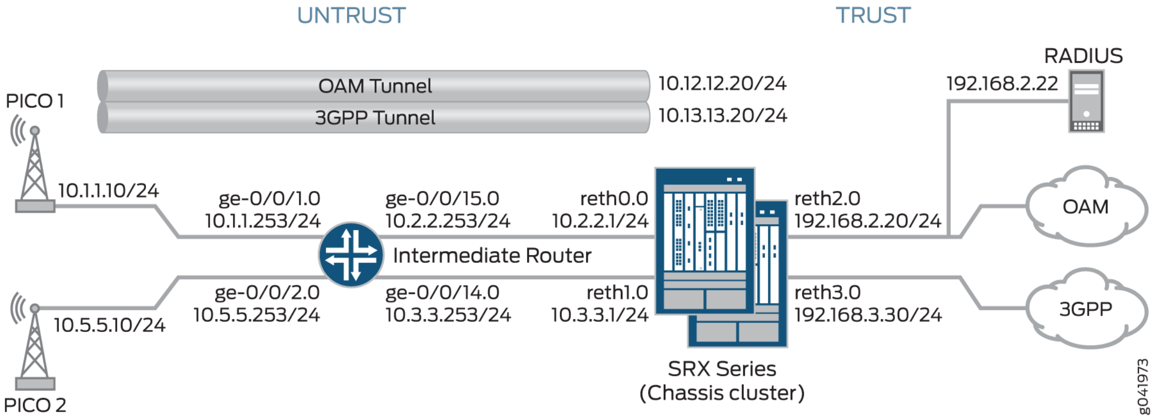
Overview

In this example, an SRX Series uses the IKEv2 configuration payload feature to propagate provisioning information to a series of pico cells. The pico cells ship from the factory with a standard configuration that allows them to connect to the SRX Series, but the pico cell provisioning information is stored on an external RADIUS server. The pico cells receive full provisioning information after establishing secure connections with provisioning servers in a protected network. IKEv2 configuration payload is supported for both IPv4 and IPV6. This example covers IKEv2 configuration payload for IPv4, however you can configure with IPv6 addresses as well.

Starting in Junos OS Release 20.3R1, we support IKEv2 IPv6 configuration payload for assigning IPv6 address on SRX5000 line running iked process. The same support is included in vSRX Virtual Firewall running iked process starting from Junos OS Release 21.1R1.

Figure 33 on page 449 shows a topology in which the SRX Series supports pico cell provisioning using the IKEv2 configuration payload feature.

Figure 33: SRX Series Support for Pico Cell Provisioning with IKEv2 Configuration Payload



Each pico cell in this topology initiates two IPsec VPNs: one for management and one for data. In this example, management traffic uses the tunnel labeled OAM Tunnel, while the data traffic flows through the tunnel labeled 3GPP Tunnel. Each tunnel supports connections with OAM and 3GPP provisioning servers on separate, configurable networks, requiring separate routing instances and VPNs. This example provides the IKE Phase 1 and Phase 2 options for establishing the OAM and 3GPP VPNs.

In this example, the SRX Series acts as the IKEv2 configuration payload server, acquiring provisioning information from the RADIUS server and providing that information to the pico cell clients. The SRX Series returns the provisioning information for each authorized client in the IKEv2 configuration payload during tunnel negotiation. The SRX Series cannot be used as a client device.

Additionally, the SRX Series uses the IKEv2 configuration payload information to update the Traffic Selector initiator (TSi) and Traffic Selector responder (TSr) values exchanged with the client during tunnel negotiation. The configuration payload uses the TSi and TSr values that are configured on the SRX Series using the `proxy-identity` statement at the [`edit security ipsec vpn` *`vpn-name`* `ike`] hierarchy level. The TSi and TSr values define the network traffic for each VPN.

The intermediate router routes pico cell traffic to the appropriate interfaces on the SRX Series.

The following process describes the connection sequence:

1. The pico cell initiates an IPsec tunnel with the SRX Series using the factory configuration.

2. The SRX Series authenticates the client using the client certificate information and the root certificate of the CA that is enrolled in the SRX Series. After authentication, the SRX Series passes the IKE identity information from the client certificate to the RADIUS server in an authorization request.

3. After authorizing the client, the RADIUS server responds to the SRX Series with the client provisioning information:

    - IP address (TSi value)

    - IP subnet mask (optional; the default is 32 bit)

    - DNS address (optional)

4. The SRX Series returns the provisioning information in the IKEv2 configuration payload for each client connection, and exchanges final TSi and TSr values with the pico cells. In this example, the SRX Series provides the following TSi and TSr information for each VPN:

VPN Connection	TSi/TSr Values Provided by SRX
Pico 1 OAM	TSi: 10.12.1.201/32, TSr: 192.168.2.0/24
Pico 1 3GPP	TSi: 10.13.1.201/32, TSr: 192.168.3.0/24, TSr: 10.13.0.0/16
Pico 2 OAM	TSi: 10.12.1.205/32, TSr: 192.168.2.0/24
Pico 2 3GPP	TSi: 10.13.1.205/32, TSr: 192.168.3.0/24, TSr: 10.13.0.0/16

    If the provisioning information supplied by the RADIUS server includes a subnet mask, the SRX Series returns a second TSr value for the client connection that includes the IP subnet. This enables intrapeer communication for devices on that subnet. In this example, intrapeer communication is enabled for the subnet associated with the 3GPP VPN (13.13.0.0/16).

    The IKEv2 configuration payload feature is supported for both point-to-multipoint secure tunnel (st0) interfaces and point-to-point interfaces. For point-to-multipoint interfaces, the interfaces must be numbered, and the addresses provided in the configuration payload must be within the subnetwork range of the associated point-to-multipoint interface.

Starting in Junos OS Release 20.1R1, we support IKEv2 configuration payload feature with point-to-point interfaces on SRX5000 line and vSRX Virtual Firewall running iked.

Table 55 on page 451 shows the Phase 1 and Phase 2 options configured on the SRX Series, including information for establishing both OAM and 3GPP tunnels.

Table 55: Phase 1 and Phase 2 Options for the SRX Series

Option	Value
IKE proposal:	
Proposal name	IKE_PROP
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	group5
Authentication algorithm	SHA-1
Encryption algorithm	AES 256 CBC
IKE policy:	
IKE Policy name	IKE_POL
Local certificate	Example_SRX
IKE gateway (OAM):	
IKE policy	IKE_POL
Remote IP address	dynamic
IKE user type	group-ike-id
Local IKE ID	hostname srx_series.example.net

Table 55: Phase 1 and Phase 2 Options for the SRX Series *(Continued)*

Option	Value
Remote IKE ID	hostname .pico_cell.net
External interface	reth0.0
Access profile	radius_pico
IKE version	v2-only
IKE gateway (3GPP):	
IKE policy	IKE_POL
Remote IP address	Dynamic
IKE user type	group-ike-id
Local IKE ID	distinguished-name wildcard OU=srx_series
Remote IKE ID	distinguished-name wildcard OU=pico_cell
External interface	reth1
Access profile	radius_pico
IKE version	v2-only
IPsec proposal:	
Proposal name	IPSEC_PROP
Protocol	ESP

Table 55: Phase 1 and Phase 2 Options for the SRX Series *(Continued)*

Option	Value
Authentication algorithm	HMAC SHA-1 96
Encryption algorithm	AES 256 CBC
IPsec policy:	
Policy name	IPSEC_POL
Perfect Forward Secrecy (PFS) keys	group5
IPsec proposals	IPSEC_PROP
IPsec VPN (OAM):	
Bind interface	st0.0
IKE gateway	OAM_GW
Local proxy-identity	192.168.2.0/24
Remote proxy-identity	0.0.0.0/0
IPsec policy	IPSEC_POL
IPsec VPN (3GPP):	
Bind interface	st0.1
IKE gateway	3GPP_GW
Local proxy-identity	192.168.3.0/24

Table 55: Phase 1 and Phase 2 Options for the SRX Series *(Continued)*

Option	Value
Remote proxy-identity	0.0.0.0/0
IPsec policy	IPSEC_POL

Certificates are stored on the pico cells and the SRX Series.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Configuration

IN THIS SECTION

Configuring the SRX Series

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set chassis cluster reth-count 5
set chassis cluster node 0
set chassis cluster node 1
set chassis cluster redundancy-group 0 node 0 priority 250
set chassis cluster redundancy-group 0 node 1 priority 150
set chassis cluster redundancy-group 1 node 0 priority 220
set chassis cluster redundancy-group 1 node 1 priority 149
set chassis cluster redundancy-group 1 interface-monitor ge-3/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-3/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-3/2/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/2/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-3/2/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/2/1 weight 255
set interfaces ge-3/0/0 gigether-options redundant-parent reth0
set interfaces ge-3/0/1 gigether-options redundant-parent reth1
set interfaces ge-3/2/0 gigether-options redundant-parent reth2
set interfaces ge-3/2/1 gigether-options redundant-parent reth3
set interfaces ge-8/0/0 gigether-options redundant-parent reth0
set interfaces ge-8/0/1 gigether-options redundant-parent reth1
set interfaces ge-8/2/0 gigether-options redundant-parent reth2
set interfaces ge-8/2/1 gigether-options redundant-parent reth3
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.2.2.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.3.3.1/24
set interfaces reth2 redundant-ether-options redundancy-group 1
set interfaces reth2 unit 0 family inet address 192.168.2.20/24
set interfaces reth3 redundant-ether-options redundancy-group 1
set interfaces reth3 unit 0 family inet address 192.168.3.20/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.12.1.20/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 10.13.1.20/24
set routing-options static route 10.1.0.0/16 next-hop 10.2.2.253
set routing-options static route 10.5.0.0/16 next-hop 10.2.2.253
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces reth0.0
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone oam-trust host-inbound-traffic system-services all
set security zones security-zone oam-trust host-inbound-traffic protocols all
set security zones security-zone oam-trust interfaces reth2.0
set security zones security-zone oam-trust interfaces st0.0
set security zones security-zone 3gpp-trust host-inbound-traffic system-services all
set security zones security-zone 3gpp-trust host-inbound-traffic protocols all
set security zones security-zone 3gpp-trust interfaces reth3.0
set security zones security-zone 3gpp-trust interfaces st0.1
set access profile radius_pico authentication-order radius
set access profile radius_pico radius-server 192.168.2.22 secret "$ABC123"
set access profile radius_pico radius-server 192.168.2.22 routing-instance VR-OAM
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate example_SRX
set security ike gateway OAM_GW ike-policy IKE_POL
set security ike gateway OAM_GW dynamic hostname .pico_cell.net
set security ike gateway OAM_GW dynamic ike-user-type group-ike-id
set security ike gateway OAM_GW local-identity hostname srx_series.example.net
set security ike gateway OAM_GW external-interface reth0.0
set security ike gateway OAM_GW aaa access-profile radius_pico
set security ike gateway OAM_GW version v2-only
set security ike gateway 3GPP_GW ike-policy IKE_POL
set security ike gateway 3GPP_GW dynamic distinguished-name wildcard OU=pico_cell
set security ike gateway 3GPP_GW dynamic ike-user-type group-ike-id
set security ike gateway 3GPP_GW local-identity distinguished-name wildcard OU=srx_series
set security ike gateway 3GPP_GW external-interface reth1.0
set security ike gateway 3GPP_GW aaa access-profile radius_pico
set security ike gateway 3GPP_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec proposal IPSEC_PROP lifetime-seconds 300
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn OAM_VPN bind-interface st0.0
set security ipsec vpn OAM_VPN ike gateway OAM_GW
set security ipsec vpn OAM_VPN ike proxy-identity local 192.168.2.0/24
set security ipsec vpn OAM_VPN ike proxy-identity remote 0.0.0.0/0
set security ipsec vpn OAM_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn 3GPP_VPN bind-interface st0.1
set security ipsec vpn 3GPP_VPN ike gateway 3GPP_GW
set security ipsec vpn 3GPP_VPN ike proxy-identity local 192.168.3.0/24
set security ipsec vpn 3GPP_VPN ike proxy-identity remote 0.0.0.0/0
set security ipsec vpn 3GPP_VPN ike ipsec-policy IPSEC_POL
set routing-instances VR-OAM instance-type virtual-router
set routing-instances VR-OAM interface reth2.0
set routing-instances VR-OAM interface st0.0
set routing-instances VR-3GPP instance-type virtual-router
set routing-instances VR-3GPP interface reth3.0
set routing-instances VR-3GPP interface st0.1
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the SRX Series:

1. Configure the chassis cluster.

```
[edit chassis cluster]
user@host# set reth-count 5
user@host# set node 0
user@host# set node 1
user@host#set redundancy-group 0 node 0 priority 250
user@host#set redundancy-group 0 node 1 priority 150
user@host#set redundancy-group 1 node 0 priority 220
user@host#set redundancy-group 1 node 1 priority 149
user@host# set redundancy-group 1 interface-monitor ge-3/0/0 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/0/0 weight 255
user@host# set redundancy-group 1 interface-monitor ge-3/0/1 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/0/1 weight 255
user@host# set redundancy-group 1 interface-monitor ge-3/2/0 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/2/0 weight 255
user@host# set redundancy-group 1 interface-monitor ge-3/2/1 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/2/1 weight 255
```

2. Configure interfaces.

```
[edit interfaces]
user@host# set ge-3/0/0 gigether-options redundant-parent reth0
user@host# set ge-3/0/1 gigether-options redundant-parent reth1
user@host# set ge-3/2/0 gigether-options redundant-parent reth2
user@host# set ge-3/2/1 gigether-options redundant-parent reth3
user@host# set ge-8/0/0 gigether-options redundant-parent reth0
user@host# set ge-8/0/1 gigether-options redundant-parent reth1
user@host# set ge-8/2/0 gigether-options redundant-parent reth2
user@host# set ge-8/2/1 gigether-options redundant-parent reth3
user@host# set reth0 redundant-ether-options redundancy-group 1
```

```
user@host# set reth0 unit 0 family inet address 10.2.2.1/24
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 10.3.3.1/24
user@host# set reth2 redundant-ether-options redundancy-group 1
user@host# set reth2 unit 0 family inet address 192.168.2.20/24
user@host# set reth3 redundant-ether-options redundancy-group 1
user@host# set reth3 unit 0 family inet address 192.169.3.20/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.12.1.20/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 10.13.1.20/24
```

3. Configure routing options.

```
[edit routing-options]
user@host# set static route 10.1.0.0/16 next-hop 10.2.2.253
user@host# set static route 10.5.0.0/16 next-hop 10.2.2.253
```

4. Specify security zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic protocols all
user@host# set host-inbound-traffic system-services all
user@host# set interfaces reth0.0
user@host# set interfaces reth1.0
[edit security zones security-zone oam-trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces reth2.0
user@host# set interfaces st0.0
[edit security zones security-zone 3gpp-trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces reth3.0
user@host# set interfaces st0.1
```

5. Create the RADIUS profile.

```
[edit access profile radius_pico]
user@host# set authentication-order radius
```

```
user@host# set radius-server 192.168.2.22 secret “$ABC123”
user@host# set radius-server 192.168.2.22 routing-instance VR-OAM
```

6. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate example_SRX
[edit security ike gateway OAM_GW]
user@host# set ike-policy IKE_POL
user@host# set dynamic hostname .pico_cell.net
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity hostname srx.example.net
user@host# set external-interface reth0.0
user@host# set aaa access-profile radius_pico
user@host# set version v2-only
[edit security ike gateway 3GPP_GW]
user@host# set ike-policy IKE_POL
user@host# set dynamic distinguished-name wildcard OU=pico_cell
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name wildcard OU=srx_series
user@host# set external-interface reth1.0
user@host# set aaa access-profile radius_pico
user@host# set version v2-only
```

7. Specify Phase 2 options.

```
[edit set security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 300
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn OAM_VPN]
```

```
user@host# set bind-interface st0.0
user@host# set ike gateway OAM_GW
user@host# set ike proxy-identity local 192.168.2.0/24
user@host# set ike proxy-identity remote 0.0.0.0/0
user@host# set ike ipsec-policy IPSEC_POL
[edit security ipsec vpn 3GPP_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway 3GPP_GW
user@host# set ike proxy-identity local 192.168.3.0/24
user@host# set ike proxy-identity remote 0.0.0.0/0
user@host# set ike ipsec-policy IPSEC_POL
```

8. Specify the routing instances.

```
[edit routing-instances VR-OAM]
user@host# set instance-type virtual router
user@host# set interface reth2.0
user@host# set interface st0.0
[edit routing-instances VR-3GPP]
user@host# set instance-type virtual router
user@host# set interface reth3.0
user@host# set interface st0.1
```

9. Specify security policies to permit site-to-site traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show chassis cluster`, `show interfaces`, `show security zones`, `show access profile radius_pico`, `show security ike`, `show security ipsec`, `show routing-instances`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis cluster
reth-count 5
node 0
node 1
redundancy-group 0{
    node 0 priority 250;
    node 1 priority 150;
    redundancy-group 1 {
    node 0 priority 220;
    node 1 priority 149;
    interface-monitor {
        ge-3/0/0 weight 255;
        ge-8/0/0 weight 255;
        ge-3/0/1 weight 255;
        ge-8/0/1 weight 255;
        ge-3/2/0 weight 255;
        ge-8/2/0 weight 255;
        ge-3/2/1 weight 255;
        ge-8/2/1 weight 255;
    }
}
[edit]
user@host# show interfaces
ge-3/0/0 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-3/0/1 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-3/2/0 {
    gigether-options {
        redundant-parent reth2;
    }
}
ge-3/2/1 {
    gigether-options {
        redundant-parent reth3;
    }
}
ge-8/0/0 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-8/0/1 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-8/2/0 {
    gigether-options {
        redundant-parent reth2;
    }
}
ge-8/2/1 {
    gigether-options {
        redundant-parent reth3;
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 10.2.2.1/24;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 10.3.3.1/24;
        }
    }
}
reth2 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.168.2.20/24;
        }
    }
}
reth3 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.168.3.20/24;
        }
    }
}
st0 {
    unit 0{
        multipoint;
        family inet {
            address 12.12.1.20/24;
        }
    }
    unit 1{
        multipoint;
        family inet {
            address 13.13.1.20/24;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.0.0/16 next-hop 10.2.2.253;
    route 10.5.0.0/16 next-hop 10.2.2.253;
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        reth1.0;
        reth0.0;
    }
}
security-zone oam-trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        reth2.0;
        st0.0;
    }
}
security-zone 3gpp-trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        reth3.0;
        st0.1;
    }
}
[edit]
user@host# show access profile radius_pico
authentication-order radius;
radius-server {
    192.168.2.22 {
        secret "$ABC123";
        routing-instance VR-OAM;
    }
```

```
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    certificate {
        local-certificate example_SRX;
    }
}
gateway OAM_GW {
    ike-policy IKE_POL;
    dynamic {
        hostname .pico_cell.net;
        ike-user-type group-ike-id;
    }
    local-identity hostname srx_series.example.net;
    external-interface reth0.0;
    aaa access-profile radius_pico;
    version v2-only;
}
gateway 3GPP_GW {
    ike-policy IKE_POL;
    dynamic {
        distinguished-name {
            wildcard OU=pico_cell;
        }
        ike-user-type group-ike-id;
    }
    local-identity distinguished-name;
    external-interface reth1.0;
    aaa access-profile radius_pico;
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 300;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals IPSEC_PROP;
}
vpn OAM_VPN {
    bind-interface st0.0;
    ike {
        gateway OAM_GW;
        proxy-identity {
            local 192.168.2.0/24;
            remote 0.0.0.0/0;
        }
        ipsec-policy IPSEC_POL;
    }
}
vpn 3GPP_VPN {
    bind-interface st0.1;
    ike {
        gateway 3GPP_GW;
        proxy-identity {
            local 192.168.3.0/24;
            remote 0.0.0.0/0;
        }
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show routing-instances
VR-OAM {
    instance-type virtual-router;
    interface reth2.0;
    interface st0.0;
}
VR-3GPP {
    instance-type virtual-router;
    interface reth3.0;
    interface st0.1;
```

```
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Intermediate Router

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.253/24
set interfaces ge-0/0/2 unit 0 family inet address 10.5.5.253/24
set interfaces ge-0/0/14 unit 0 family inet address 10.3.3.253/24
set interfaces ge-0/0/15 unit 0 family inet address 10.2.2.253/24
set routing-options static route 192.168.3.0/24 next-hop 10.2.2.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/14.0
set security zones security-zone trust interfaces ge-0/0/15.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/2.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the intermediate router:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.1.1.253/24
user@host# set ge-0/0/2 unit 0 family inet address 10.5.5.253/24
user@host# set ge-0/0/14 unit 0 family inet address 10.3.3.253/24
user@host# set ge-0/0/15 unit 0 family inet address 10.2.2.253/24
```

2. Configure routing options.

```
[edit routing-options]
user@host# set static route 192.168.3.0/24  next-hop 10.2.2.1
```

3. Specify security zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic protocols all
user@host# set host-inbound-traffic system-services all
user@host# set interfaces ge-0/0/14.0
user@host# set interfaces ge-0/0/15.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces ge-0/0/2.0
```

4. Specify security policies.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.253/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.5.5.253/24;
        }
    }
}
ge-0/0/14 {
    unit 0 {
        family inet {
            address 10.3.3.253/24;
        }
    }
}
ge-0/0/15 {
    unit 0 {
        family inet {
            address 10.2.2.253/24;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 192.168.3.0/24 next-hop 10.2.2.1;
}
[edit]
```

```
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/14.0;
        ge-0/0/15.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
        ge-0/0/2.0;
    }
}
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Pico Cell (Sample Configuration)

Step-by-Step Procedure

The pico cell information in this example is provided for reference. Detailed pico cell configuration information is beyond the scope of this document. The pico cell factory configuration must include the following information:

- Local certificate (X.509v3) and IKE identity information
- Traffic Selector (TSi, TSr) values set to any/any (0.0.0.0/0)
- SRX Series IKE identity information and public IP address
- Phase 1 and Phase 2 proposals that match the SRX Series configuration

The pico cells in this example use strongSwan open source software for IPsec-based VPN connections. This information is used by the SRX Series for pico cell provisioning using the IKEv2 configuration payload feature. In networks where many devices are being deployed, the pico cell configuration can be identical except for the certificate (leftcert) and identity (leftid) information. The following sample configurations illustrate factory settings.

1. Review the Pico 1 configuration:

   Pico 1: Sample Configuration

```
conn %default
        ikelifetime=8h
        keylife=1h
        rekeymargin=1m
        keyingtries=1
        keyexchange=ikev2
        authby=pubkey
        mobike=no

conn oam
        left=%any
        leftsourceip=%config
        leftcert=/usr/local/etc/ipsec.d/certs/<cert_name>
        leftid=pico1.pico_cell.net
        leftfirewall=yes
        reauth=yes
        right=10.2.2.1/24
        rightid=srx_series.example.net
        rightsubnet=0.0.0.0/0 #peer net for proxy id
        ike=aes256-sha-modp1536!
        esp=aes256-sha-modp1536!
        auto=add

conn 3gpp
        left=%any
        leftsourceip=%config
        leftcert=/usr/local/etc/ipsec.d/certs/<cert_name>
        leftid=”C=US, ST=CA, L=Sunnyvale, O=org, OU=pico_cell, CN=pico1”
        leftfirewall=yes
        reauth=yes
        right=10.3.3.1/24
        rightid=”OU=srx_series”
        rightsubnet=0.0.0.0/0 #peer net for proxy id
        ike=aes256-sha-modp1536!
        esp=aes256-sha-modp1536!
        auto=add
```

2. Review the Pico 2 configuration:

Pico 2 Sample Configuration

```
conn %default
        ikelifetime=8h
        keylife=1h
        rekeymargin=1m
        keyingtries=1
        keyexchange=ikev2
        authby=pubkey
        mobike=no

conn oam
        left=%any
        leftsourceip=%config
        leftcert=/usr/local/etc/ipsec.d/certs/<cert_name>
        leftid=pico2.pico_cell.net
        leftfirewall=yes
        #reauth=no
        right=10.2.2.1/24
        rightid=srx_series.example.net
        rightsubnet=0.0.0.0/0 #peer net for proxy id
        ike=aes256-sha-modp1536!
        esp=aes256-sha-modp1536!
        auto=add

conn 3gpp
        left=%any
        leftsourceip=%config
        leftcert=/usr/local/etc/ipsec.d/certs/<cert_name>
        leftid=”C=US, ST=CA, L=Sunnyvale, O=org, OU=pico_cell, CN=pico2”
        leftfirewall=yes
        #reauth=no
        right=10.3.3.1/24
        rightid=”OU=srx_series”
        rightsubnet=0.0.0.0/0 #peer net for proxy id
        ike=aes256-sha-modp1536!
        esp=aes256-sha-modp1536!
        auto=add
```

Configuring the RADIUS Server (Sample Configuration using a FreeRADIUS)

Step-by-Step Procedure

The RADIUS server information in this example is provided for reference. Complete RADIUS server configuration information is beyond the scope of this document. The following information is returned to the SRX Series by the RADIUS server:

- Framed-IP-Address
- Framed-IP-Netmask (optional)
- Primary-DNS and Secondary-DNS (optional)

In this example, the RADIUS server has separate provisioning information for the OAM and 3GPP connections. The User-Name is taken from the client certificate information provided in the SRX Series authorization request.

If the RADIUS server acquires client provisioning information from a DHCP server, the client identity information relayed to the DHCP server by the RADIUS server must be consistent with the client IKE identity information relayed to the RADIUS server by the SRX Series Firewall. This ensures the continuity of the client identity across the various protocols.

The communication channel between the SRX Series Firewall and the RADIUS server is protected by a RADIUS shared secret.

1. Review the RADIUS configuration for the Pico 1 OAM VPN. The RADIUS server has the following information:

   Sample RADIUS configuration in Junos OS Releases 12.3X48 and Junos OS releases prior to 15.1X49-D160, 17.3R3, 17.4R2, 18.1R3, 18.2R2, 18.3R1, and 18.1R3-S2:

   FreeRADIUS configuration example:

   ```
   DEFAULT User-Name =~ "device@example.net", Cleartext-Password := "juniper"
           Service-Type = Framed-User,
           Framed-IP-Address = 10.12.1.201,
           Framed-IP-Netmask = 255.255.255.255,
           Primary-Dns = 192.168.2.104,
           Secondary-Dns = 192.168.2.106,
   ```

   Sample RADIUS configuration starting from Junos OS Releases 15.X49-D161, 15.1X49-D170, 17.3R3, 17.4R2, 18.1R3, 18.2R2, 18.3R1, and 18.1R3-S2:

   FreeRADIUS configuration example:

   ```
   DEFAULT User-Name =~ "device@example.net", Auth-Type := "Accept"
           Service-Type = Framed-User,
           Framed-IP-Address = 10.12.1.201,
           Framed-IP-Netmask = 255.255.255.255,
           Primary-Dns = 192.168.2.104,
           Secondary-Dns = 192.168.2.106,
   ```

   In this case, the RADIUS server provides the default subnet mask (255.255.255.255), which blocks intrapeer traffic.

2. Review the RADIUS configuration for the Pico 1 3GPP VPN. The RADIUS server has the following information:

   Sample RADIUS configuration in Junos OS Releases 12.3X48 and Junos OS releases prior to 15.1X49-D160, 17.3R3, 17.4R2, 18.1R3, 18.2R2, 18.3R1, and 18.1R3-S2:

   FreeRADIUS configuration example:

   ```
   DEFAULT User-Name =~ "device@example.net", Cleartext-Password := "juniper"
          Service-Type = Framed-User,
           Framed-IP-Address = 10.13.1.201.10,
           Framed-IP-Netmask = 255.255.0.0,
   ```

```
          Primary-Dns = 192.168.2.104,
          Secondary-Dns = 192.168.2.106,
```

Sample RADIUS configuration starting from Junos OS Releases 15.X49-D161, 15.1X49-D170, 17.3R3, 17.4R2, 18.1R3, 18.2R2, 18.3R1, and 18.1R3-S2:

FreeRADIUS configuration example:

```
DEFAULT User-Name =~ "device@example.net", Auth-Type := "Accept"
        Service-Type = Framed-User,
         Framed-IP-Address = 10.13.1.201.10,
         Framed-IP-Netmask = 255.255.0.0,
         Primary-Dns = 192.168.2.104,
         Secondary-Dns = 192.168.2.106,
```

In this case, the RADIUS server provides a subnet mask value (255.255.0.0), which enables intrapeer traffic.

Starting in Junos OS Release 20.1R1, you can configure a common password for IKEv2 configuration payload requests for an IKE gateway configuration. The common password in the range of 1 to 128 characters allows the administrator to define a common password. This password is used between the SRX Series Firewall and the RADIUS server when the SRX Series Firewall requesting an IP address on behalf of a remote IPsec peer using IKEv2 configuration payload. RADIUS server validate the credentials before it provides any IP information to the SRX Series Firewall for the configuration payload request. You can configure the common password using `config-payload-password` *`configured-password`* configuration statement at `[edit security ike gateway gateway-name aaa access-profile access-profile-name]` hierarchy level. Additionally, this example creates two tunnels from the same client certificate by using different parts of the certificate for User-Name (IKE identity) information.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the IKE Phase 1 Status for the SRX Series

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode on node 0, enter the **show security ike security-associations** command. After obtaining an index number from the command, use the **show security ike security-associations detail** command.

```
user@host# show security ike security-associations
node0:
--------------------------------------------------------------------------
Index      State  Initiator cookie  Responder cookie  Mode  Remote Address
553329718  UP     99919a471d1a5278  3be7c5a49172e6c2  IKEv2 10.1.1.1
1643848758 UP     9e31d4323195a195  4d142438106d4273  IKEv2 10.1.1.1
```

```
user@host# show security ike security-associations index 553329718 detail
node0:
--------------------------------------------------------------------------
IKE peer 10.1.1.1, Index 553329718, Gateway Name: OAM_GW
  Location: FPC 2, PIC 0, KMD-Instance 1
  Role: Responder, State: UP
  Initiator cookie: 99919a471d1a5278, Responder cookie: 3be7c5a49172e6c2
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 10.2.2.1:500, Remote: 10.1.1.1:500
  Lifetime: Expires in 28738 seconds
  Peer ike-id: C=US, ST=CA, L=Sunnyvale, O=org, OU=pico_cell, CN=pico1
  aaa assigned IP: 10.12.1.201
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2104
   Output bytes  :                  425
  Input  packets:                    2
```

```
     Output packets:                    1
    IPSec security associations: 0 created, 0 deleted
    Phase 2 negotiations in progress: 1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs with pico cells devices. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. This example shows only the IKE Phase 1 SA for the OAM VPN; however, a separate IKE Phase 1 SA will be displayed showing the IKE Phase 1 parameters for the 3GPP VPN.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA: you can use the `show security ike security-associations index detail` command to get more information about the SA.
- Remote address—Verify that the local IP address is correct and that port 500 is being used for peer-to-peer communication.
- Role responder state:
  - Up—The Phase 1 SA has been established.
  - Down—There was a problem establishing the Phase 1 SA.
- Peer (remote) IKE ID—Verify the certificate information is correct.
- Local identity and remote identity—Verify these addresses are correct.
- Mode—Verify that the correct mode is being used.

Verify that the following items are correct in your configuration:

- External interfaces (the interface must be the one that sends IKE packets)
- IKE policy parameters
- Phase 1 proposal parameters (must match between peers)

The `show security ike security-associations` command lists the following additional information about security associations:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)

- Role information

  Troubleshooting is best performed on the peer using the responder role.

- Initiator and responder information

- Number of IPsec SAs created

- Number of Phase 2 negotiations in progress

Verifying IPsec Security Associations for the SRX Series

Purpose

Verify the IPsec status.

Action

From operational mode on node 0, enter the **show security ipsec security-associations** command. After obtaining an index number from the command, use the **show security ipsec security-associations detail** command.

```
user@host# show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 2
  ID         Algorithm             SPI     Life:sec/kb Mon lsys Port Gateway
  <214171651 ESP:aes-cbc-256/sha1 cc2869e2 3529/        -  root 500  10.1.1.1
  >214171651 ESP:aes-cbc-256/sha1 c0a54936 3529/        -  root 500  10.1.1.1
  <205520899 ESP:aes-cbc-256/sha1 84e49026 3521/        -  root 500  10.1.1.1
  >205520899 ESP:aes-cbc-256/sha1 c4ed1849 3521/        -  root 500  10.1.1.1
```

```
user@host# show security ipsec security-associations detail
node0:
--------------------------------------------------------------------------
Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x604a29
Last Tunnel Down Reason: SA not initiated
  ID: 214171651 Virtual-system: root, VPN Name: 3GPP_VPN
  Local Gateway: 10.3.3.1, Remote Gateway: 10.1.1.1
  Local Identity: list(any:0,ipv4_subnet(any:0-65535,[0..7]=192.168.3.0/24),
ipv4_subnet(any:0-65535,[0..7]=10.13.0.0/16))
  Remote Identity: ipv4(any:0,[0..3]=10.13.1.201)
  DF-bit: clear
  Bind-interface: st0.1

Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
Last Tunnel Down Reason: SA not initiated
  Location: FPC 6, PIC 0, KMD-Instance 2
  Direction: inbound, SPI: cc2869e2, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3523 seconds
  Lifesize Remaining:
  Soft lifetime: Expires in 2965 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64

  Location: FPC 6, PIC 0, KMD-Instance 2
  Direction: outbound, SPI: c0a54936, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3523 seconds
  Lifesize Remaining:
  Soft lifetime: Expires in 2965 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64

ID: 205520899 Virtual-system: root, VPN Name: OAM_VPN
Local Gateway: 10.2.2.1, Remote Gateway: 10.1.1.1
Local Identity: ipv4_subnet(any:0-65535,[0..7]=192.168.2.0/24)
Remote Identity: ipv4(any:0,[0..3]=10.12.1.201)
Version: IKEv2
  DF-bit: clear
  Bind-interface: st0.0

Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
Last Tunnel Down Reason: SA not initiated
  Location: FPC 2, PIC 0, KMD-Instance 1
  Direction: inbound, SPI: 84e49026, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3515 seconds
  Lifesize Remaining:
  Soft lifetime: Expires in 2933 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
```

```
    Anti-replay service: counter-based enabled, Replay window size: 64

    Location: FPC 2, PIC 0, KMD-Instance 1
    Direction: outbound, SPI: c4ed1849, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3515 seconds
    Lifesize Remaining:
    Soft lifetime: Expires in 2933 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

This examples shows the active IKE Phase 2 SAs for Pico 1. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IPsec policy parameters in your configuration. For each Phase 2 SA (OAM and 3GPP), information is provided in both the inbound and outboard direction. The output from the `show security ipsec security-associations` command lists the following information:

- The remote gateway has an IP address of 10.1.1.1.

- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3529/ value indicates that the Phase 2 lifetime expires in 3529 seconds, and that no lifesize has been specified, which indicates that it is unlimited. The Phase 2 lifetime can differ from the Phase 1 lifetime, because Phase 2 is not dependent on Phase 1 after the VPN is up.

- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.

- The virtual system (vsys) is the root system, and it always lists 0.

The above output from the `show security ipsec security-associations index` *`index_id`* `detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.

- Authentication and encryption algorithms used.

- Phase 2 proposal parameters (must match between peers).
- Secure tunnel (st0.0 and st0.1) bindings to the OAM and 3GPP gateways.

SEE ALSO

IPsec Overview | 20

Introduction to PKI in Junos OS | 31

IKE Policy with a Trusted CA

This example shows how to bind a trusted CA server to an IKE policy of the peer.

Before you begin, you must have a list of all the trusted CAs you want to associate with the IKE policy of the peer.

You can associate an IKE policy to a single trusted CA profile or a trusted CA group. For establishing a secure connection, the IKE gateway uses the IKE policy to limit itself to the configured group of CAs (ca-profiles) while validating the certificate. A certificate issued by any source other than the trusted CA or trusted CA group is not validated. If there is a certificate validation request coming from an IKE policy then the associated CA profile of the IKE policy will validate the certificate. If an IKE policy is not associated with any CA then by default the certificate is validated by any one of the configured CA profiles.

In this example, a CA profile named `root-ca` is created and a `root-ca-identity` is associated to the profile.

You can configure a maximum of 20 CA profiles that you want to add to a trusted CA group. You cannot commit your configuration if you configure more than 20 CA profiles in a trusted CA group.

1. Create a CA profile and associate a CA identifier to the profile.

```
[edit]
user@host# set security pki ca-profile root-ca ca-identity root-ca
```

2. Define an IKE proposal and the IKE proposal authentication method.

```
[edit]
user@host# set security ike proposal ike_prop authentication-method rsa-signatures
```

3. Define the Diffie-Hellman group, authentication algorithm, an encryption algorithm for the IKE proposal.

   ```
   [edit]
   user@host# set security ike proposal ike_prop dh-group group2
   user@host# set security ike proposal ike_prop authentication-algorithm sha-256
   user@host# set security ike proposal ike_prop encryption-algorithm aes-256-cbc
   ```

4. Configure an IKE policy and associate the policy with the IKE proposal.

   ```
   [edit]
   user@host# set security ike policy ike_policy proposals ike_prop
   ```

5. Configure a local certificate identifier for the IKE policy.

   ```
   [edit]
   user@host# set security ike policy ike_policy certificate local-certificate SPOKE
   ```

6. Define the CA to be used for the IKE policy.

   ```
   [edit]
   user@host# set security ike policy ike_policy certificate trusted-ca ca-profile root-ca
   ```

To view the CA profiles and the trusted CA groups configured on your device, run `show security pki` command.

```
user@host# show security ike
    proposal ike_prop {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
}
policy ike_policy {
    proposals ike_prop;
    certificate {
        local-certificate SPOKE;
        trusted-ca ca-profile root-ca;
```

```
    }
}
```

The `show security ike` command displays the CA profile group under the IKE policy named `ike_policy` and the certificate associated with the IKE policy.

SEE ALSO

Understanding Certificate Authority Profiles | 46

Secure Tunnel Interface in a Virtual Router

IN THIS SECTION

A secure tunnel interface (st0) is an internal interface that is used by route-based VPNs to route cleartext traffic to an IPsec VPN tunnel.

Understanding Virtual Router Support for Route-Based VPNs

IN THIS SECTION

This feature includes routing-instance support for route-based VPNs. In previous releases, when an st0 interface was put in a nondefault routing instance, the VPN tunnels on this interface did not work properly. In the Junos OS 10.4 release, the support is enabled to place st0 interfaces in a routing instance, where each unit is configured in point-to-point mode or multipoint mode. Therefore, VPN traffic now works correctly in a nondefault VR. You can now configure different subunits of the st0 interface in different routing instances. The following functions are supported for nondefault routing instances:

- Manual key management
- Transit traffic
- Self-traffic
- VPN monitoring
- Hub-and-spoke VPNs
- Encapsulating Security Payload (ESP) protocol
- Authentication Header (AH) protocol
- Aggressive mode or main mode
- st0 anchored on the loopback (lo0) interface
- Maximum number of virtual routers (VRs) supported on an SRX Series Firewall
- Applications such as Application Layer Gateway (ALG), Intrusion Detection and Prevention (IDP), and Content Security
- Dead peer detection (DPD)
- *Chassis cluster* active/backup
- Open Shortest Path First (OSPF) over st0
- Routing Information Protocol (RIP) over st0
- Policy-based VPN inside VR

Understanding Virtual Router Limitations

When you configure VPN on SRX Series Firewalls, overlapping of IP addresses across virtual routers is supported with the following limitations:

- An IKE external interface address cannot overlap with any other virtual router.
- An internal or trust interface address can overlap across any other virtual router.
- An st0 interface address cannot overlap in route-based VPN in point-to-multipoint tunnels such as NHTB.

- An st0 interface address can overlap in route-based VPN in point-to-point tunnels.

SEE ALSO

IPsec Overview | 20

Example: Configuring an st0 Interface in a Virtual Router

IN THIS SECTION

This example shows how to configure an st0 interface in a virtual router.

Requirements

Before you begin, configure the interfaces and assign the interfaces to security zones. See "*Security Zones Overview*".

Overview

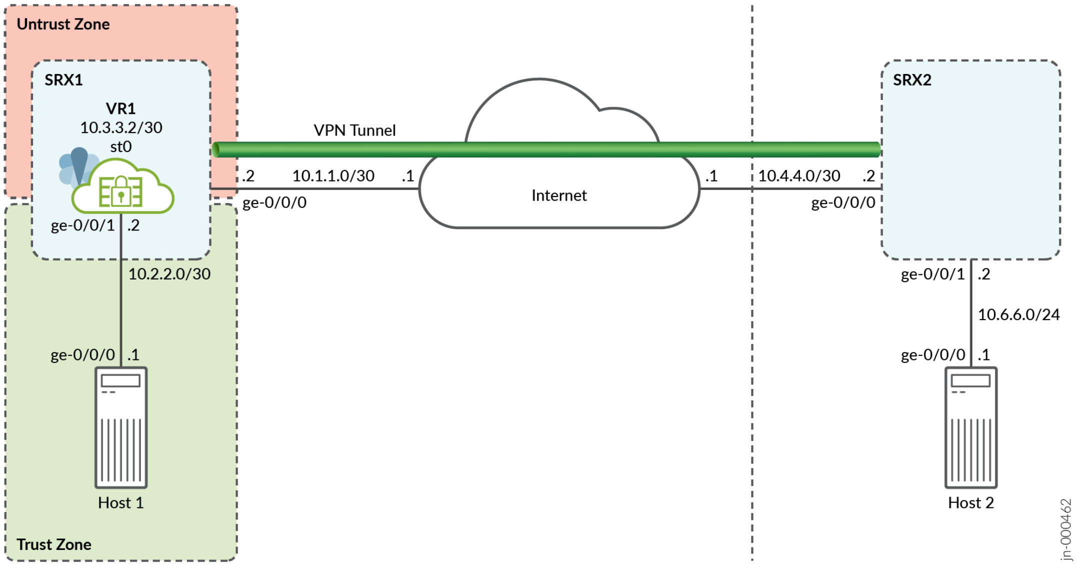
In this example, you perform the following operations:

- Configure the interfaces.
- Configure IKE Phase 1 proposals.
- Configure IKE policies, and reference the proposals.
- Configure an IKE gateway, and reference the policy.
- Configure Phase 2 proposals.
- Configure policies, and reference the proposals.
- Configure AutoKey IKE, and reference the policy and gateway.

- Configure the security policy.
- Configure the routing instance.
- Configure the VPN bind to tunnel interface.
- Configure the routing options.

Figure 34 on page 486 shows the topology used in this example.

Figure 34: Secure Tunnel Interface in a Virtual Router



Following tables show the configuration parameters.

Table 56: Interface, Routing Instance, Static Route, and Security Zone Information for SRX1

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	10.1.1.2/30
	ge-0/0/1.0	10.2.2.2/30
	st0.0 (tunnel interface)	10.3.3.2/30

Table 56: Interface, Routing Instance, Static Route, and Security Zone Information for SRX1 *(Continued)*

Feature	Name	Configuration Parameters
Routing instance (Virtual Router)	VR1	ge-0/0/1.0 st0.0
Static routes	10.6.6.0/24	The next hop is st0.0.
Security zones	trust	• The ge-0/0/1 interface is bound to this zone.
	untrust	• The ge-0/0/0 interface is bound to this zone. • The st0.0 interface is bound to this zone.

Table 57: IKE Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	first_ikeprop	• Authentication method: pre-shared-keys
Policy	first_ikepol	• Mode: main • Proposal reference: first_ikeprop • IKE policy authentication method: pre-shared-keys
Gateway	first	• IKE policy reference: first_ikepol • External interface: ge-0/0/0.0 • Gateway address: 10.4.4.2

Table 58: IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	first_ipsecprop	• protocol: esp • authentication-algorithm: hmac-md5-96 • encryption-algorithm: 3des-cbc
Policy	first_ipsecpol	• IPsec proposal reference: first_ipsecprop
VPN	first_vpn	• IKE gateway reference: first • IPsec policy reference: first_ipsecpol • Bind to interface: st0.0 • establish-tunnels immediately

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.2/30
set interfaces ge-0/0/1 unit 0 family inet address 10.2.2.2/30
set interfaces st0 unit 0 family inet address 10.3.3.2/30
set security zones security-zone trust interfaces ge-0/0/1
set security zones security-zone untrust interfaces ge-0/0/0
set security zones security-zone untrust interfaces st0.0
set security ike proposal first_ikeprop authentication-method pre-shared-keys
set security ike proposal first_ikeprop dh-group group2
set security ike proposal first_ikeprop authentication-algorithm md5
set security ike proposal first_ikeprop encryption-algorithm 3des-cbc
set security ike policy first_ikepol mode main
set security ike policy first_ikepol proposals first_ikeprop
set security ike policy first_ikepol pre-shared-key ascii-text "$ABC123"
set security ike gateway first ike-policy first_ikepol
set security ike gateway first address 10.4.4.2
set security ike gateway first external-interface ge-0/0/0.0
set security ipsec proposal first_ipsecprop protocol esp
set security ipsec proposal first_ipsecprop authentication-algorithm hmac-md5-96
set security ipsec proposal first_ipsecprop encryption-algorithm 3des-cbc
set security ipsec policy first_ipsecpol perfect-forward-secrecy keys group1
set security ipsec policy first_ipsecpol proposals first_ipsecprop
set security ipsec vpn first_vpn bind-interface st0.0
set security ipsec vpn first_vpn ike gateway first
set security ipsec vpn first_vpn ike ipsec-policy first_ipsecpol
set security ipsec vpn first_vpn establish-tunnels immediately
set security policies from-zone trust to-zone untrust policy p1 match source-address any
set security policies from-zone trust to-zone untrust policy p1 match destination-address any
set security policies from-zone trust to-zone untrust policy p1 match application any
set security policies from-zone trust to-zone untrust policy p1 then permit
set security policies from-zone untrust to-zone trust policy p2 match source-address any
set security policies from-zone untrust to-zone trust policy p2 match destination-address any
set security policies from-zone untrust to-zone trust policy p2 match application any
set security policies from-zone untrust to-zone trust policy p2 then permit
set routing-instances VR1 instance-type virtual-router
set routing-instances VR1 interface ge-0/0/1.0
set routing-instances VR1 interface st0.0
set routing-instances VR1 routing-options static route 10.6.6.0/24 next-hop st0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure an st0 in a VR:

1. Configure the interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.2/30
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.2.2.2/30
user@host# set interfaces st0 unit 0 family inet address 10.3.3.2/30
```

2. Configure security zones.

```
[edit]
user@host# set security zones security-zone trust interfaces ge-0/0/1
user@host# set security zones security-zone untrust interfaces ge-0/0/0
user@host# set security zones security-zone untrust interfaces st0.0
```

3. Configure Phase 1 of the IPsec tunnel.

```
[edit security ike]
user@host# set proposal first_ikeprop authentication-method pre-shared-keys
user@host# set proposal first_ikeprop dh-group group2
user@host# set proposal first_ikeprop authentication-algorithm md5
user@host# set proposal first_ikeprop encryption-algorithm 3des-cbc
```

4. Configure the IKE policies, and reference the proposals.

```
[edit security ike]
user@host# set policy first_ikepol mode main
user@host# set policy first_ikepol proposals first_ikeprop
user@host# set policy first_ikepol pre-shared-key ascii-text "$ABC123"
```

5. Configure the IKE gateway, and reference the policy.

```
[edit security ike]
user@host# set gateway first ike-policy first_ikepol
user@host# set gateway first address 10.4.4.2
user@host# set gateway first external-interface ge-0/0/0.0
```

6. Configure Phase 2 of the IPsec tunnel.

```
[edit security ipsec]
user@host# set proposal first_ipsecprop protocol esp
user@host# set proposal first_ipsecprop authentication-algorithm hmac-md5-96
user@host# set proposal first_ipsecprop encryption-algorithm 3des-cbc
```

7. Configure the policies, and reference the proposals.

```
[edit security ipsec]
user@host# set policy first_ipsecpol perfect-forward-secrecy keys group1
user@host# set policy first_ipsecpol proposals first_ipsecprop
```

8. Configure AutoKey IKE, and reference the policy and gateway.

```
[edit security ipsec]
user@host# set vpn first_vpn ike gateway first
user@host# set vpn first_vpn ike ipsec-policy first_ipsecpol
user@host# set vpn first_vpn establish-tunnels immediately
```

9. Configure the VPN bind to tunnel interface.

```
[edit security ipsec]
user@host# set vpn first_vpn bind-interface st0.0
```

10. Configure the security policy.

```
[edit security policies]
user@host# set from-zone trust to-zone untrust policy p1 match source-address any
user@host# set from-zone trust to-zone untrust policy p1 match destination-address any
user@host# set from-zone trust to-zone untrust policy p1 match application any
user@host# set from-zone trust to-zone untrust policy p1 then permit
user@host# set from-zone untrust to-zone trust policy p2 match source-address any
user@host# set from-zone untrust to-zone trust policy p2 match destination-address any
user@host# set from-zone untrust to-zone trust policy p2 match application any
user@host# set from-zone untrust to-zone trust policy p2 then permit
```

11. Configure the st0 in the routing instance.

```
[edit routing-instances]
user@host# set VR1 instance-type virtual-router
user@host# set VR1 interface ge-0/0/1.0
user@host# set VR1 interface st0.0
```

12. Configure the routing options.

```
[edit routing-instances VR1 routing-options]
user@host# set static route 10.6.6.0/24 next-hop st0.0
```

Results

From configuration mode, confirm your configuration by entering the `show security` and `show routing-instances` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security
    ike {
    proposal first_ikeprop {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm md5;
        encryption-algorithm 3des-cbc;
    }
    policy first_ikepol {
        mode main;
        proposals first_ikeprop;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
    }
    gateway first {
        ike-policy first_ikepol;
        address 10.4.4.2;
        external-interface ge-0/0/0.0;
    }
}
    ipsec {
        proposal first_ipsecprop {
            protocol esp;
            authentication-algorithm hmac-md5-96;
            encryption-algorithm 3des-cbc;
        }
        policy first_ipsecpol {
            perfect-forward-secrecy {
                keys group1;
            }
            proposals first_ipsecprop;
        }
        vpn first_vpn {
            bind-interface st0.0;
            ike {
                gateway first;
                ipsec-policy first_ipsecpol;
            }
            establish-tunnels immediately;
        }
    }
policies {
    from-zone trust to-zone untrust {
        policy p1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
        from-zone untrust to-zone trust {
            policy p2 {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
zones {
    security-zone trust {
        interfaces {
            ge-0/0/1.0;
        }
    }
    security-zone untrust {
        interfaces {
            ge-0/0/0.0;
            st0.0;
        }
    }
}

user@host# show routing-instances
        VR1 {
            instance-type virtual-router;
            interface ge-0/0/1.0;
            interface st0.0;
            routing-options {
            static {
            route 10.6.6.0/24 next-hop st0.0;
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform this task:

Verifying an st0 interface in the Virtual Router

Purpose

Verify the st0 interface in the virtual router.

Action

From operational mode, enter the `show interfaces st0.0 detail` command. The number listed for routing table corresponds to the order that the routing tables in the `show route all` command.

SEE ALSO

Understanding Virtual Router Support for Route-Based VPNs | **483**

RELATED DOCUMENTATION

Route-Based IPsec VPNs | **394**

Dual Stack Tunnels over an External Interface

IN THIS SECTION

Dual-stack tunnels—parallel IPv4 and IPv6 tunnels over a single physical interface to a peer—are supported for route-based site-to-site VPNs. A physical interface configured with both IPv4 and IPv6 addresses can be used as an external interface for IPv4 and IPv6 gateways on the same peer or on different peers at the same time.

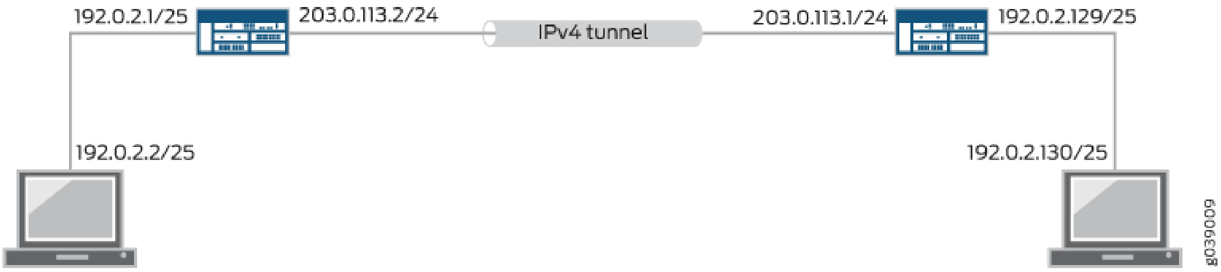
Understanding VPN Tunnel Modes

IN THIS SECTION

In VPN tunnel mode, IPsec encapsulates the original IP datagram—including the original IP header—within a second IP datagram. The outer IP header contains the IP address of the gateway, while the inner header contains the ultimate source and destination IP addresses. The outer and inner IP headers can have a protocol field of IPv4 or IPv6. SRX Series Firewalls support four tunnel modes for route-based site-to-site VPNs.

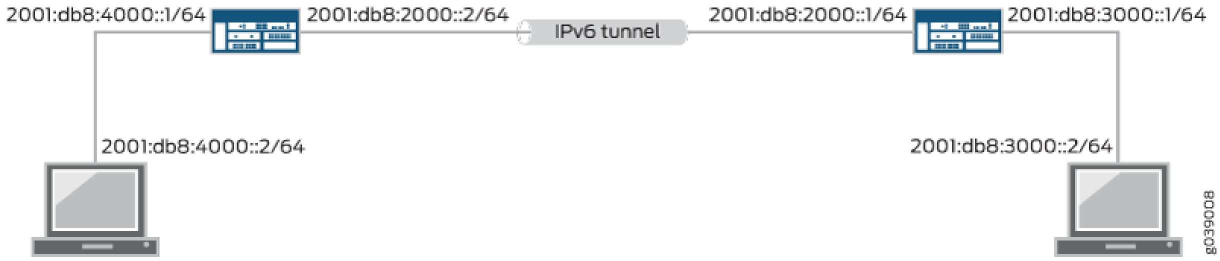
IPv4-in-IPv4 tunnels encapsulate IPv4 packets inside IPv4 packets, as shown in Figure 35 on page 496. The protocol fields for both the outer and the inner headers are IPv4.

Figure 35: IPv4-in-IPv4 Tunnel



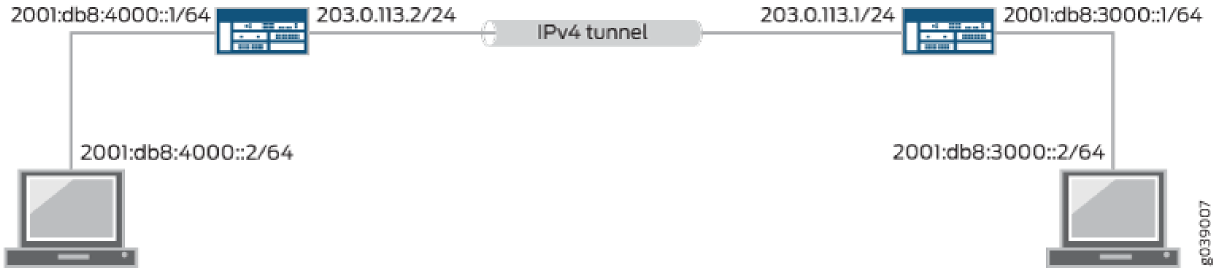
IPv6-in-IPv6 tunnels encapsulate IPv6 packets inside IPv6 packets, as shown in Figure 36 on page 497. The protocol fields for both the outer and inner headers are IPv6.

Figure 36: IPv6-in-IPv6 Tunnel



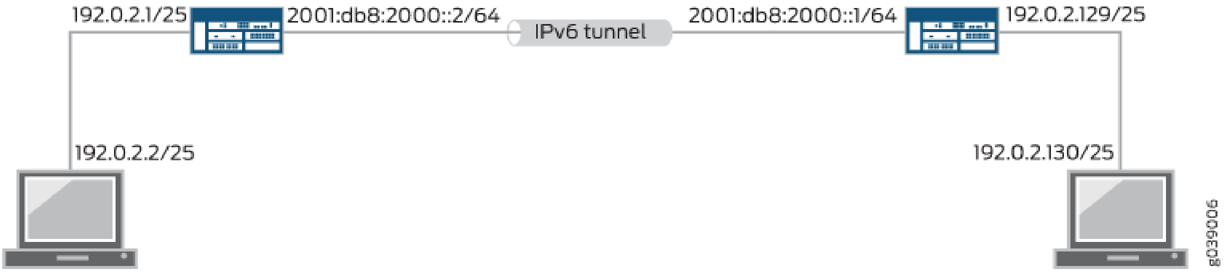
IPv6-in-IPv4 tunnels encapsulate IPv6 packets inside IPv4 packets, as shown in Figure 37 on page 497. The protocol field for the outer header is IPv4 and the protocol field for the inner header is IPv6.

Figure 37: IPv6-in-IPv4 Tunnel



IPv4-in-IPv6 tunnels encapsulate IPv4 packets inside IPv6 packets, as shown in Figure 38 on page 497. The protocol field for the outer header is IPv6 and the protocol field for the inner header is IPv4.

Figure 38: IPv4-in-IPv6 Tunnel



A single IPsec VPN tunnel can carry both IPv4 and IPv6 traffic. For example, an IPv4 tunnel can operate in both IPv4-in-IPv4 and IPv6-in-IPv4 tunnel modes at the same time. To allow both IPv4 and IPv6 traffic over a single IPsec VPN tunnel, the st0 interface bound to that tunnel must be configured with both `family inet` and `family inet6`.

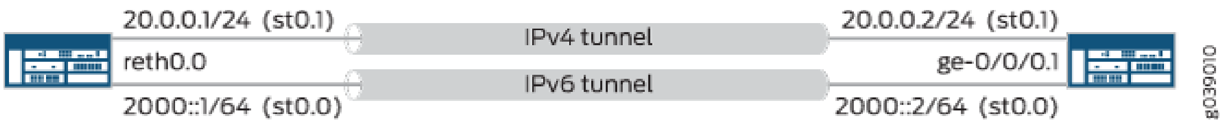
A physical interface configured with both IPv4 and IPv6 addresses can be used as the external interface for parallel IPv4 and IPv6 tunnels to a peer in a route-based site-to-site VPN. This feature is known as *dual-stack tunnels* and requires separate st0 interfaces for each tunnel.

For policy-based VPNs, IPv6-in-IPv6 is the only tunnel mode supported and it is only supported on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

Understanding Dual-Stack Tunnels over an External Interface

Dual-stack tunnels—parallel IPv4 and IPv6 tunnels over a single physical interface to a peer—are supported for route-based site-to-site VPNs. A physical interface configured with both IPv4 and IPv6 addresses can be used as the external interface to IPv4 and IPv6 gateways on the same peer or on different peers at the same time. In Figure 39 on page 498, the physical interfaces reth0.0 and ge-0/0/0.1 support parallel IPv4 and IPv6 tunnels between two devices.

Figure 39: Dual-Stack Tunnels



In Figure 39 on page 498, separate secure tunnel (st0) interfaces must be configured for each IPsec VPN tunnel. Parallel IPv4 and IPv6 tunnels that are bound to the same st0 interface are not supported.

A single IPsec VPN tunnel can carry both IPv4 and IPv6 traffic. For example, an IPv4 tunnel can operate in both IPv4-in-IPv4 and IPv6-in-IPv4 tunnel modes at the same time. To allow both IPv4 and IPv6 traffic over a single IPsec VPN tunnel, the st0 interface bound to that tunnel must be configured with both `family inet` and `family inet6`.

If multiple addresses in the same address family are configured on the same external interface to a VPN peer, we recommend that you configure `local-address` at the [`edit security ike gateway` *`gateway-name`*] hierarchy level.

If `local-address` is configured, the specified IPv4 or IPv6 address is used as the local gateway address. If only one IPv4 and one IPv6 address is configured on a physical external interface, `local-address` configuration is not required.

The local-address value must be an IP address that is configured on an interface on the SRX Series Firewall. We recommend that local-address belong to the external interface of the IKE gateway. If local-address does not belong to the external interface of the IKE gateway, the interface must be in the same zone as the external interface of the IKE gateway and an intra-zone security policy must be configured to permit traffic.

The local-address value and the remote IKE gateway address must be in the same address family, either IPv4 or IPv6.

If local-address is not configured, the local gateway address is based on the remote gateway address. If the remote gateway address is an IPv4 address, the local gateway address is the primary IPv4 address of the external physical interface. If the remote gateway address is an IPv6 address, the local gateway address is the primary IPv6 address of the external physical interface.

SEE ALSO

VPN Feature Support for IPv6 Addresses | 350

Understanding IPv6 IKE and IPsec Packet Processing | 356

VPN Feature Support for IPv6 Addresses | 350

Example: Configuring Dual-Stack Tunnels over an External Interface

IN THIS SECTION

This example shows how to configure parallel IPv4 and IPv6 tunnels over a single external physical interface to a peer for route-based site-to-site VPNs.

Requirements

Before you begin, read "Understanding VPN Tunnel Modes" on page 496.

The configuration shown in this example is only supported with route-based site-to-site VPNs.

Overview

IN THIS SECTION

In this example, a redundant Ethernet interface on the local device supports parallel IPv4 and IPv6 tunnels to a peer device:

- The IPv4 tunnel carries IPv6 traffic; it operates in IPv6-in-IPv4 tunnel mode. The secure tunnel interface st0.0 bound to the IPv4 tunnel is configured with family inet6 only.
- The IPv6 tunnel carries both IPv4 and IPv6 traffic; it operates in both IPv4-in-IPv6 and IPv6-in-IPv6 tunnel modes. The secure tunnel interface st0.1 bound to the IPv6 tunnel is configured with both family inet and family inet6.

Table 59 on page 500 shows the Phase 1 options used in this example. The Phase 1 option configuration includes two IKE gateway configurations, one to the IPv6 peer and the other to the IPv4 peer.

Table 59: Phase 1 Options for Dual-Stack Tunnel Configuration

Option	Value
IKE proposal	ike_proposal
Authentication method	Preshared keys
Authentication algorithm	MD5
Encryption algorithm	3DES CBC
Lifetime	3600 seconds
IKE policy	ike_policy

Table 59: Phase 1 Options for Dual-Stack Tunnel Configuration *(Continued)*

Option	Value
Mode	Aggressive
IKE proposal	ike_proposal
Preshared key	ASCII text
IPv6 IKE gateway	ike_gw_v6
IKE policy	ike_policy
Gateway address	2000::2
External interface	reth1.0
IKE version	IKEv2
IPv4 IKE gateway	ike_gw_v4
IKE policy	ike_policy
Gateway address	20.0.0.2
External interface	reth1.0

Table 60 on page 502 shows the Phase 2 options used in this example. The Phase 2 option configuration includes two VPN configurations, one for the IPv6 tunnel and the other for the IPv4 tunnel.

Table 60: Phase 2 Options for Dual-Stack Tunnel Configuration

Option	Value
IPsec proposal	ipsec_proposal
Protocol	ESP
Authentication algorithm	HMAC SHA-1 96
Encryption algorithm	3DES CBC
IPsec policy	ipsec_policy
Proposal	ipsec_proposal
IPv6 VPN	test_s2s_v6
Bind interface	st0.1
IKE gateway	ike_gw_v6
IKE IPsec policy	ipsec_policy
Establish tunnels	Immediately
IPv4 VPN	test_s2s_v4
Bind interface	st0.0
IKE gateway	ike_gw_4
IKE IPsec policy	ipsec_policy

The following static routes are configured in the IPv6 routing table:

- Route IPv6 traffic to 3000::1/128 through st0.0.
- Route IPv6 traffic to 3000::2/128 through st0.1.

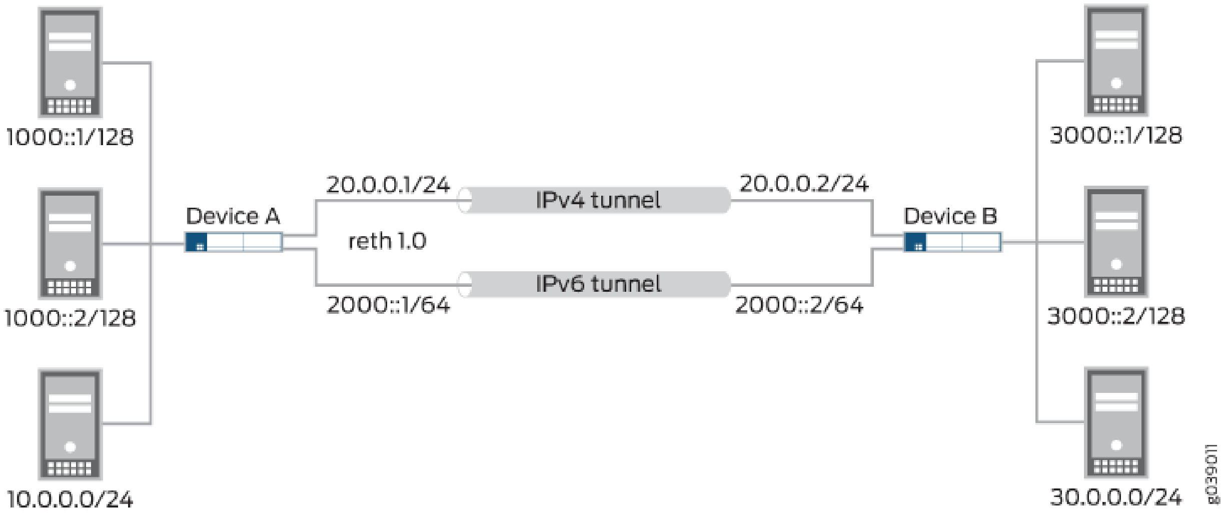
A static route is configured in the default (IPv4) routing table to route IPv4 traffic to 30.0.0.0/24 through st0.1.

Flow-based processing of IPv6 traffic must be enabled with the `mode flow-based` configuration option at the `[edit security forwarding-options family inet6]` hierarchy level.

Topology

In Figure 40 on page 503, the SRX Series Firewall A supports IPv4 and IPv6 tunnels to device B. IPv6 traffic to 3000::1/128 is routed through the IPv4 tunnel, while IPv6 traffic to 3000::2/128 and IPv4 traffic to 30.0.0.0/24 are routed through the IPv6 tunnel.

Figure 40: Dual-Stack Tunnel Example



Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 gigether-options redundant-parent reth1
set interfaces ge-8/0/1 gigether-options redundant-parent reth1
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 20.0.0.1/24
set interfaces reth1 unit 0 family inet6 address 2000::1/64
set interfaces st0 unit 0 family inet6
set interfaces st0 unit 1 family inet
set interfaces st0 unit 1 family inet6
set security ike proposal ike_proposal authentication-method pre-shared-keys
set security ike proposal ike_proposal authentication-algorithm md5
set security ike proposal ike_proposal encryption-algorithm 3des-cbc
set security ike proposal ike_proposal lifetime-seconds 3600
set security ike policy ike_policy mode aggressive
set security ike policy ike_policy proposals ike_proposal
set security ike policy ike_policy pre-shared-key ascii-text "$ABC123"
set security ike gateway ike_gw_v6 ike-policy ike_policy
set security ike gateway ike_gw_v6 address 2000::2
set security ike gateway ike_gw_v6 external-interface reth1.0
set security ike gateway ike_gw_v6 version v2-only
set security ike gateway ike_gw_v4 ike-policy ike_policy
set security ike gateway ike_gw_v4 address 20.0.0.2
set security ike gateway ike_gw_v4 external-interface reth1.0
set security ipsec proposal ipsec_proposal protocol esp
set security ipsec proposal ipsec_proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_proposal encryption-algorithm 3des-cbc
set security ipsec policy ipsec_policy proposals ipsec_proposal
set security ipsec vpn test_s2s_v6 bind-interface st0.1
set security ipsec vpn test_s2s_v6 ike gateway ike_gw_v6
set security ipsec vpn test_s2s_v6 ike ipsec-policy ipsec_policy
set security ipsec vpn test_s2s_v6 establish-tunnels immediately
set security ipsec vpn test_s2s_v4 bind-interface st0.0
set security ipsec vpn test_s2s_v4 ike gateway ike_gw_v4
set security ipsec vpn test_s2s_v4 ike ipsec-policy ipsec_policy
set routing-options rib inet6.0 static route 3000::1/128 next-hop st0.0
set routing-options rib inet6.0 static route 3000::2/128 next-hop st0.1
set routing-options static route 30.0.0.0/24 next-hop st0.1
set security forwarding-options family inet6 mode flow-based
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure dual-stack tunnels:

1. Configure the external interface.

   ```
   [edit interfaces]
   user@host# set ge-0/0/1 gigether-options redundant-parent reth1
   user@host# set ge-8/0/1 gigether-options redundant-parent reth1
   user@host# set reth1 redundant-ether-options redundancy-group 1
   user@host# set reth1 unit 0 family inet address 20.0.0.1/24
   user@host# set reth1 unit 0 family inet6 address 2000::1/64
   ```

2. Configure the secure tunnel interfaces.

   ```
   [edit interfaces]
   user@host# set st0 unit 0 family inet6
   user@host# set st0 unit 1 family inet
   user@host# set st0 unit 1 family inet6
   ```

3. Configure Phase 1 options.

   ```
   [edit security ike proposal ike_proposal]
   user@host# set authentication-method pre-shared-keys
   user@host# set authentication-algorithm md5
   user@host# set encryption-algorithm 3des-cbc
   user@host# set lifetime-seconds 3600
   [edit security ike policy ike_policy]
   user@host# set mode aggressive
   user@host# set proposals ike_proposal
   user@host# set pre-shared-key ascii-text "$ABC123"
   [edit security ike gateway ike_gw_v6]
   ```

```
user@host# set ike-policy ike_policy
user@host# set address 2000::2
user@host# set external-interface reth1.0
user@host# set version v2-only
[edit security ike gateway ike_gw_v4]
user@host# set ike-policy ike_policy
user@host# set address 20.0.0.2
user@host# set external-interface reth1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec_proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm 3des-cbc
[edit security ipsec policy ipsec_policy]
user@host# set proposals ipsec_proposal
[edit security ipsec vpn test_s2s_v6 ]
user@host# set bind-interface st0.1
user@host# set ike gateway ike_gw_v6
user@host# set ike ipsec-policy ipsec_policy
user@host# set establish-tunnels immediately
[edit security ipsec vpn test_s2s_v4]
user@host# set bind-interface st0.0
user@host# set ike gateway ike_gw_v4
user@host# set ike ipsec-policy ipsec_policy
```

5. Configure static routes.

```
[edit routing-options rib inet6.0]
user@host# set static route 3000::1/128 next-hop st0.0
user@host# set static route 3000::2/128 next-hop st0.1
[edit routing-options]
user@host# set static route 30.0.0.0/24 next-hop st0.1
```

6. Enable IPv6 flow-based forwarding.

```
[edit security forwarding-options]
user@host# set family inet6 mode flow-based
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike`, `show security ipsec`, `show routing-options`, and `show security forwarding-options` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
    user@host# show interfaces
    ge-0/0/1 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-8/0/1 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 20.0.0.1/24;
            }
            family inet6 {
                address 2000::1/64;
            }
        }
    }
    st0 {
        unit 0 {
            family inet;
            family inet6;
        }
        unit 1 {
            family inet6;
        }
    }
    [edit]
    user@host# show security ike
```

```
proposal ike_proposal {
    authentication-method pre-shared-keys;
    authentication-algorithm md5;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 3600;
}
policy ike_policy {
    mode aggressive;
    proposals ike_proposal;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway ike_gw_v6 {
    ike-policy ike_policy;
    address 2000::2;
    external-interface reth1.0;
    version v2-only;
}
gateway ike_gw_4 {
    ike-policy ike_policy;
    address 20.0.0.2;
    external-interface reth1.0;
}
[edit]
user@host# show security ipsec
proposal ipsec_proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
}
policy ipsec_policy {
    proposals ipsec_proposal;
}
vpn test_s2s_v6 {
    bind-interface st0.1;
    ike {
        gateway ike_gw_v6;
        ipsec-policy ipsec_policy;
    }
    establish-tunnels immediately;
}
vpn test_s2s_v4 {
    bind-interface st0.0;
    ike {
            gateway ike_gw_4;
            ipsec-policy ipsec_policy;
        }
    }
    [edit]
    user@host# show routing-options
    rib inet6.0 {
        static {
            route 3000::1/128 next-hop st0.0;
            route 3000::2/128 next-hop st0.1;
        }
    }
    static {
        route 30.0.0.0/24 next-hop st0.1;
    }
    [edit]
user@host# show security forwarding-options
    family {
        inet6 {
            mode flow-based;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the `show security ike security-associations` command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
1081812113 UP  51d9e6df8a929624  7bc15bb40781a902  IKEv2          2000::2
1887118424 UP  d80b55b949b54f0a  b75ecc815529ae8f  Aggressive     20.0.0.2
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the peer devices.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ipsec security-associations` command.

```
user@host> show security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131074 ESP:3des/sha1 8828bd36 3571/  unlim     -   root 500   20.0.0.2
  >131074 ESP:3des/sha1 c968afd8 3571/  unlim     -   root 500   20.0.0.2
  <131073 ESP:3des/sha1 8e9e695a 3551/  unlim     -   root 500   2000::2
  >131073 ESP:3des/sha1 b3a254d1 3551/  unlim     -   root 500   2000::2
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the peer devices.

Verifying Routes

Purpose

Verify active routes.

Action

From operational mode, enter the `show route` command.

```
user@host> show route
inet.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.5.0.0/16        *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.10.0.0/16       *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.150.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.150.48.0/21     *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.155.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.157.64.0/19     *[Direct/0] 3d 01:43:23
                    > via fxp0.0
10.157.72.36/32    *[Local/0] 3d 01:43:23
                      Local via fxp0.0
10.204.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.206.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
10.209.0.0/16      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
20.0.0.0/24        *[Direct/0] 03:45:41
```

```
                    > via reth1.0
20.0.0.1/32        *[Local/0] 03:45:41
                      Local via reth1.0
30.0.0.0/24        *[Static/5] 00:07:49
                    > via st0.1
50.0.0.0/24        *[Direct/0] 03:45:42
                    > via reth0.0
50.0.0.1/32        *[Local/0] 03:45:42
                      Local via reth0.0
172.16.0.0/12      *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
192.168.0.0/16     *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
192.168.102.0/23   *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
207.17.136.0/24    *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0
207.17.136.192/32  *[Static/5] 3d 01:43:23
                    > to 10.157.64.1 via fxp0.0

inet6.0: 10 destinations, 14 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2000::/64          *[Direct/0] 03:45:41
                    > via reth1.0
2000::1/128        *[Local/0] 03:45:41
                      Local via reth1.0
3000::1/128        *[Static/5] 00:03:45
                    > via st0.0
3000::2/128        *[Static/5] 00:03:45
                    > via st0.1
5000::/64          *[Direct/0] 03:45:42
                    > via reth0.0
5000::1/128        *[Local/0] 03:45:42
                      Local via reth0.0
fe80::/64          *[Direct/0] 03:45:42
                    > via reth0.0
                    [Direct/0] 03:45:41
                    > via reth1.0
                    [Direct/0] 03:45:41
                    > via st0.0
                    [Direct/0] 03:45:13
                    > via st0.1
```

```
fe80::210:dbff:feff:1000/128
                   *[Local/0] 03:45:42
                      Local via reth0.0
fe80::210:dbff:feff:1001/128
                   *[Local/0] 03:45:41
                      Local via reth1.0
```

Meaning

The `show route` command lists active entries in the routing tables.

RELATED DOCUMENTATION

IPv6 IPsec VPNs | **349**

IPsec VPN Overview | **165**

IPsec VPN Tunnels with Chassis Clusters

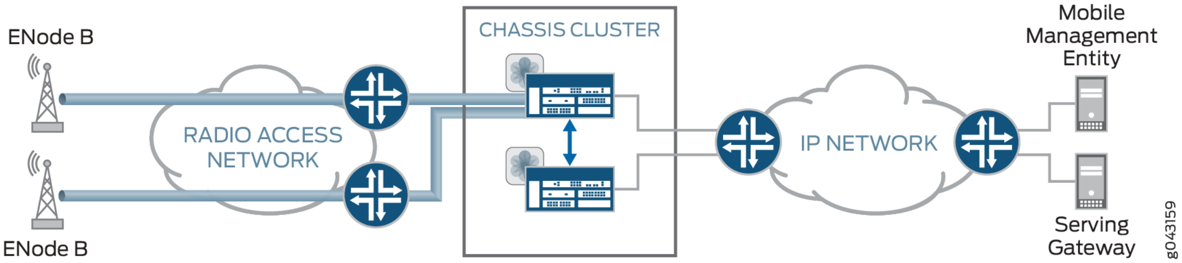
IN THIS SECTION

SRX Series Firewall support IPsec VPN tunnels in a chassis cluster setup. In an active/passive chassis cluster, all VPN tunnels terminate on the same node. In an active/active chassis cluster, VPN tunnels can terminate on either node.

Understanding Dual Active-Backup IPsec VPN Chassis Clusters

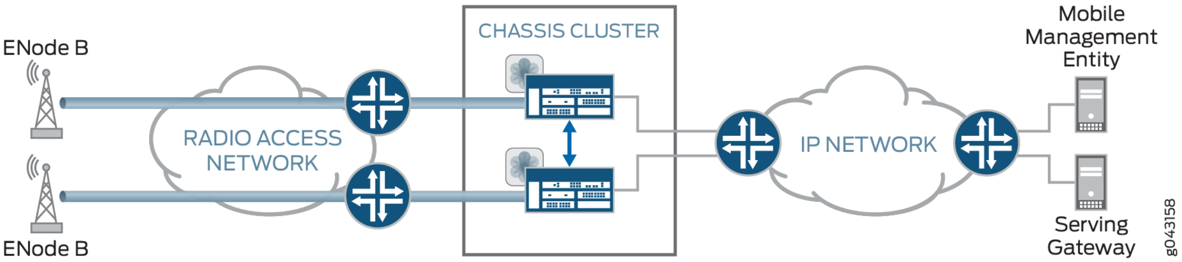
In an active/passive chassis cluster, all VPN tunnels terminate on the same node, as shown in Figure 41 on page 514.

Figure 41: Active/Passive Chassis Cluster with IPsec VPN Tunnels



In an active/active chassis cluster, VPN tunnels can terminate on either node. Both nodes in the chassis cluster can actively pass traffic through VPN tunnels on both nodes at the same time, as shown in Figure 42 on page 514. This deployment is known as *dual active-backup IPsec VPN chassis clusters*.

Figure 42: Dual Active-Backup IPsec VPN Chassis Clusters



The following features are supported with dual active-backup IPsec VPN chassis clusters:

- Route-based VPNs only. Policy-based VPNs are not supported.
- IKEv1 and IKEv2.
- Digital certificate or preshared key authentication.
- IKE and secure tunnel interfaces (st0) in virtual routers.
- Network Address Translation-Traversal (NAT-T).
- VPN monitoring.
- Dead peer detection.
- In-service software upgrade (ISSU).

- Insertion of Services Processing Cards (SPCs) on a chassis cluster device without disrupting the traffic on the existing VPN tunnels. See "VPN Support for Inserting Services Processing Cards" on page 175.
- Dynamic routing protocols.
- Secure tunnel interfaces (st0) configured in point-to-multipoint mode.
- AutoVPN with st0 interfaces in point-to-point mode with traffic selectors.
- IPv4-in-IPv4, IPv6-in-IPv4, IPv6-in-IPv6 and IPv4-in-IPv6 tunnel modes.
- Fragmented traffic.
- The loopback interface can be configured as the external interface for the VPN.

Dual active-backup IPsec VPN chassis clusters cannot be configured with Z-mode flows. Z-mode flows occur when traffic enters an interface on a chassis cluster node, passes through the fabric link, and exits through an interface on the other cluster node.

SEE ALSO

Chassis Cluster User Guide for SRX Series Devices

Example: Configuring Redundancy Groups for Loopback Interfaces

IN THIS SECTION

This example shows how to configure a redundancy group (RG) for a loopback interface in order to prevent VPN failure. Redundancy groups are used to bundle interfaces into a group for failover purpose in a chassis cluster setup.

Requirements

This example uses the following hardware and software:

- A pair of supported chassis cluster SRX Series Firewall
- An SSG140 device or equivalent
- Two switches
- Junos OS Release 12.1x44-D10 or later for SRX Series Firewall

Before you begin:

Understand chassis cluster redundant Ethernet interfaces. See Chassis Cluster User Guide for SRX Series Devices.

Overview

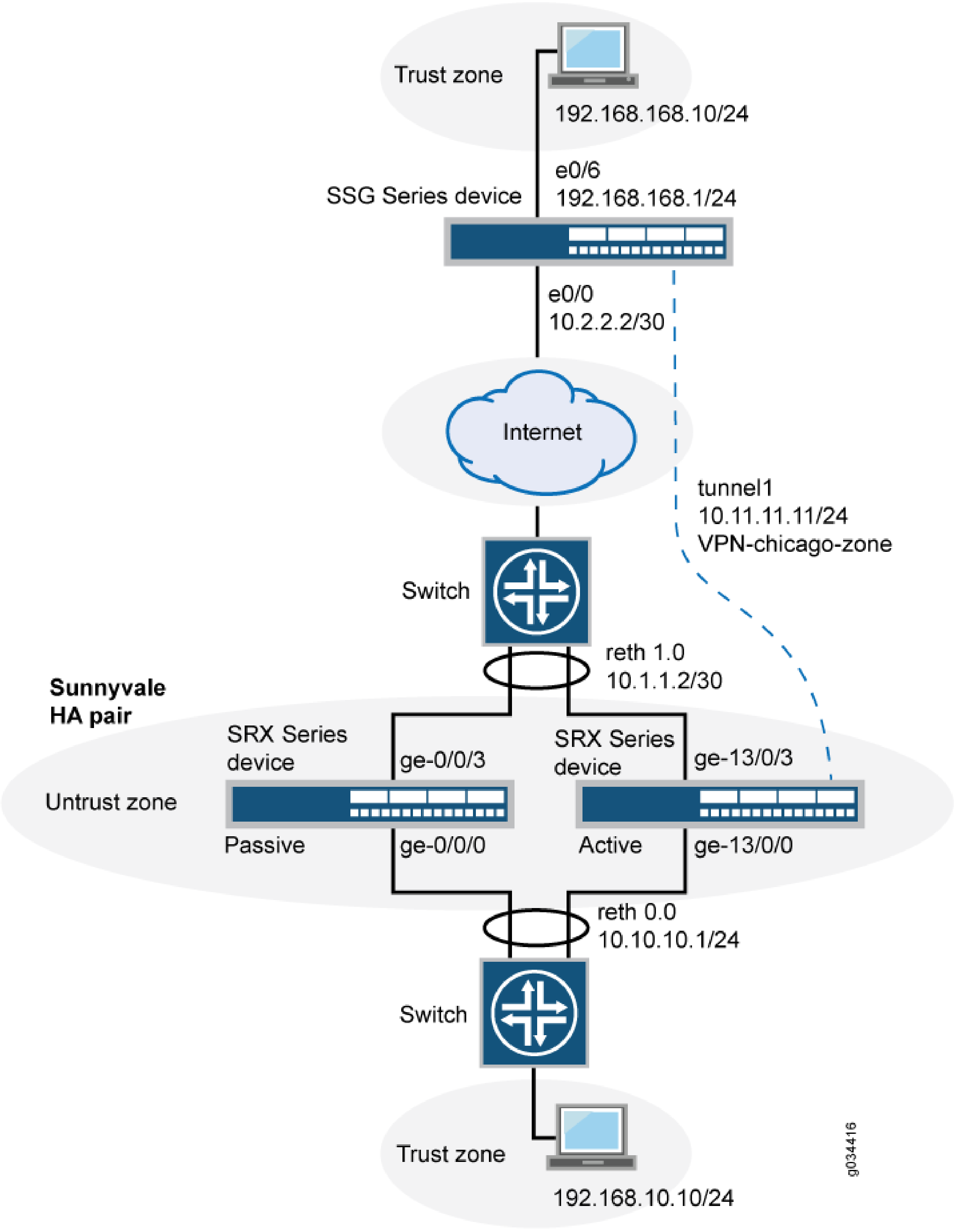
An Internet Key Exchange (IKE) gateway needs an external interface to communicate with a peer device. In a chassis cluster setup, the node on which the external interface is active selects a Services Processing Unit (SPU) to support the VPN tunnel. IKE and IPsec packets are processed on that SPU. Therefore, the active external interface decides the anchor SPU.

In a chassis cluster setup, the external interface is a redundant Ethernet interface. A redundant Ethernet interface can go down when its physical (child) interfaces are down. You can configure a loopback interface as an alternative physical interface to reach the peer gateway. Loopback interfaces can be configured on any redundancy group. This redundancy group configuration is only checked for VPN packets, because only VPN packets must find the anchor SPU through the active interface.

You must configure lo0.x in a custom virtual router, since lo0.0 is in the default virtual router and only one loopback interface is allowed in a virtual router.

Figure 43 on page 517 shows an example of a loopback chassis cluster VPN topology. In this topology, the SRX Series Firewall chassis cluster device is located in Sunnyvale, California. The SRX Series Firewall chassis cluster device works as a single gateway in this setup. The SSG Series device (or a third-party device) is located in Chicago, Illinois. This device acts as a peer device to the SRX chassis cluster and it helps to build a VPN tunnel.

Figure 43: Loopback Interface for Chassis Cluster VPN



lo0.1 10.3.3.3/30 Untrust zone	A logical interface on loopback that may be active on either node in the cluster depending on the activeness of its RG.
st0.0 10.11.11.10/24 vpn-chicago-zone	A logical interface on the secure tunnel interface for IPSec VPN tunnel.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces lo0 redundant-pseudo-interface-options redundancy-group 1
set interfaces lo0 unit 1 family inet address 10.3.3.3/30
set routing-instances vr1 instance-type virtual-router
set routing-instances vr1 interface lo0.1
set routing-instances vr1 interface reth0.0
set routing-instances vr1 interface reth1.0
set routing-instances vr1 interface st0.0
set routing-instances vr1 routing-options static route 192.168.168.1/24 next-hop st0.0
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposal-set standard
set security ike policy ike-policy1 pre-shared-key ascii-text "$ABC123"
set security ike gateway t-ike-gate ike-policy ike-policy1
set security ike gateway t-ike-gate address 10.2.2.2
set security ike gateway t-ike-gate external-interface lo0.1
set security ipsec proposal p2-std-p1 authentication-algorithm hmac-sha1-96
set security ipsec proposal p2-std-p1 encryption-algorithm 3des-cbc
set security ipsec proposal p2-std-p1 lifetime-seconds 180
set security ipsec proposal p2-std-p2 authentication-algorithm hmac-sha1-96
set security ipsec proposal p2-std-p2 encryption-algorithm aes-128-cbc
set security ipsec proposal p2-std-p2 lifetime-seconds 180
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group2
set security ipsec policy vpn-policy1 proposals p2-std-p1
set security ipsec policy vpn-policy1 proposals p2-std-p2
set security ipsec vpn t-ike-vpn bind-interface st0.0
set security ipsec vpn t-ike-vpn ike gateway t-ike-gate
set security ipsec vpn t-ike-vpn ike proxy-identity local 10.10.10.1/24
set security ipsec vpn t-ike-vpn ike proxy-identity remote 192.168.168.1/24
set security ipsec vpn t-ike-vpn ike ipsec-policy vpn-policy1
```

Step-by-Step Procedure

To configure a redundancy group for a loopback interface:

1. Configure the loopback interface in one redundancy group.

   ```
   [edit interfaces]
   user@host# set lo0 redundant-pseudo-interface-options redundancy-group 1
   ```

2. Configure the IP address for the loopback interface.

   ```
   [edit interfaces]
   user@host# set lo0 unit 1 family inet address 10.3.3.3/30
   ```

3. Configure routing options.

   ```
   [edit routing-instances]
   user@host# set vr1 instance-type virtual-router
   user@host# set vr1 interface lo0.1
   user@host# set vr1 interface reth0.0
   user@host# set vr1 interface reth1.0
   user@host# set vr1 interface st0.0
   user@host# set vr1 routing-options static route 192.168.168.1/24 next-hop st0.0
   ```

4. Configure the loopback interface as an external interface for the IKE gateway.

   ```
   [edit security ike]
   user@host# set policy ike-policy1 mode main
   user@host# set policy ike-policy1 proposal-set standard
   user@host# set policy ike-policy1 pre-shared-key ascii-text "$ABC123"
   user@host# set gateway t-ike-gate ike-policy ike-policy1
   user@host# set gateway t-ike-gate address 10.2.2.2
   user@host# set gateway t-ike-gate external-interface lo0.1
   ```

5. Configure an IPsec proposal.

```
[edit security ipsec]
user@host# set proposal p2-std-p1 authentication-algorithm hmac-sha1-96
user@host# set proposal p2-std-p1 encryption-algorithm 3des-cbc
user@host# set proposal p2-std-p1 lifetime-seconds 180
user@host# set proposal p2-std-p2 authentication-algorithm hmac-sha1-96
user@host# set proposal p2-std-p2 encryption-algorithm aes-128-cbc
user@host# set proposal p2-std-p2 lifetime-seconds 180
user@host# set policy vpn-policy1 perfect-forward-secrecy keys group2
user@host# set policy vpn-policy1 proposals p2-std-p1
user@host# set policy vpn-policy1 proposals p2-std-p2
user@host# set vpn t-ike-vpn bind-interface st0.0
user@host# set vpn t-ike-vpn ike gateway t-ike-gate
user@host# set vpn t-ike-vpn ike proxy-identity local 10.10.10.1/24
user@host# set vpn t-ike-vpn ike proxy-identity remote 192.168.168.1/24
user@host# set vpn t-ike-vpn ike ipsec-policy vpn-policy1
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces lo0`, `show routing-instances`, `show security ike`, and `show security ipsec` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces lo0
        unit 1 {
            family inet {
                address 10.3.3.3/30;
            }
        }
        redundant-pseudo-interface-options {
            redundancy-group 1;
        }
```

```
[edit]
user@host# show routing-instances
    vr1 {
        instance-type virtual-router;
        interface lo0.1;
        interface reth0.0;
        interface reth1.0;
        interface st0.0;
        routing-options {
            static {
                route 192.168.168.1/24 next-hop st0.0;
            }
        }
    }
```

```
[edit]
user@host# show security ike
    policy ike-policy1 {
        mode main;
        proposal-set standard;
        pre-shared-key ascii-text "$ABC123";
    }
        gateway t-ike-gate {
            ike-policy ike-policy1;
            address 10.2.2.2;
            external-interface lo0.1;
        }
```

```
[edit]
user@host# show security ipsec
    proposal p2-std-p1 {
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm 3des-cbc;
        lifetime-seconds 180;
    }
        proposal p2-std-p2 {
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm aes-128-cbc;
            lifetime-seconds 180;
        }
        policy vpn-policy1 {
            perfect-forward-secrecy {
                keys group2;
            }
            proposals [ p2-std-p1 p2-std-p2 ];
        }
    policy vpn-policy2 {
        perfect-forward-secrecy {
            keys group2;
        }
        proposals [ p2-std-p1 p2-std-p2 ];
    }
        vpn t-ike-vpn {
            bind-interface st0.0;
            ike {
                gateway t-ike-gate;
                proxy-identity {
                    local 10.10.10.1/24;
                    remote 192.168.168.1/24;
                }
                ipsec-policy vpn-policy1;
            }
        }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Verifying the Configuration

Purpose

Verify that the configuration for redundancy groups for loopback interfaces is correct.

Action

From operational mode, enter the **show chassis cluster interfaces** command.

```
user@host> show chassis cluster interfaces
Control link status: Up
    Control interfaces:
     Index  Interface   Status
     0       em0              Up
     1       em1             Down
    Fabric link status: Up
    Fabric interfaces:
    Name    Child-interface    Status
     fab0    ge-0/0/7           Up    / Up
     fab0
     fab1    ge-13/0/7          Up    / Up
     fab1
    Redundant-ethernet Information:
    Name      Status    Redundancy-group
     reth0        Up          1
     reth1        Up          1
     reth2        Up          1
     reth3        Down      Not configured
     reth4        Down      Not configured
    Redundant-pseudo-interface Information:
    Name    Status      Redundancy-group
     lo0          Up          1
```

Meaning

The **show chassis cluster interfaces** command displays the chassis cluster interfaces information. If the status of the Redundant-pseudo-interface Information field shows the lo0 interface as Up and the status of the Redundant-ethernet Information field shows reth0, reth1, and reth2 fields as Up then your configuration is correct.

SEE ALSO

Understanding the Loopback Interface for a High Availability VPN | **1381**

RELATED DOCUMENTATION

Chassis Cluster User Guide for SRX Series Devices

Traffic Selectors in Route-Based VPNs

IN THIS SECTION

A traffic selector is an agreement between IKE peers to permit traffic through a VPN tunnel if the traffic matches a specified pair of local and remote addresses. Only the traffic that conforms to a traffic selector is permitted through the associated security association (SA).

Understanding Traffic Selectors in Route-Based VPNs

IN THIS SECTION

A traffic selector is an agreement between IKE peers to permit traffic through a tunnel if the traffic matches a specified pair of local and remote addresses. With this feature, you can define a traffic selector within a specific route-based VPN, which can result in multiple Phase 2 IPsec security associations (SAs). Only traffic that conforms to a traffic selector is permitted through the associated SA.

Starting with Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, traffic selectors can be configured with IKEv1 site-to-site VPNs. Starting with Junos OS Release 15.1X49-D100, traffic selectors can be configured with IKEv2 site-to-site VPNs.

Traffic Selector Configuration

To configure a traffic selector, use the `traffic-selector` configuration statement at the `[edit security ipsec vpn` *`vpn-name`*`]` hierarchy level. The traffic selector is defined with the mandatory `local-ip` *`ip-address/netmask`* and `remote-ip` *`ip-address/netmask`* statements. The CLI operational command `show security ipsec security-association detail` displays traffic selector information for SAs. The `show security ipsec security-association traffic-selector` *`traffic-selector-name`* CLI command displays information for a specified traffic selector.

For a given traffic selector, a single address and netmask is specified for the local and remote addresses. Traffic selectors can be configured with IPv4 or IPv6 addresses. Address books cannot be used to specify local or remote addresses.

Multiple traffic selectors can be configured for the same VPN. A maximum of 200 traffic selectors can be configured for each VPN. Traffic selectors can be used with IPv4-in-IPv4, IPv4-in-IPv6, IPv6-in-IPv6, or IPv6-in-IPv4 tunnel modes.

Below features are not supported with traffic selectors:

- VPN monitoring
- Different address families configured for the local and remote IP addresses in a traffic selector
- A remote address of 0.0.0.0/0 (IPv4) or 0::0 (IPv6) for site-to-site VPNs

  Starting with Junos OS Release 15.1X49-D140, on all SRX Series Firewalls and vSRX Virtual Firewall instances, when you configure the traffic-selector with a remote address of 0::0 (IPv6), the following **“error: configuration check-out failed”** message is displayed when performing the commit and the configuration checkout fails.
- Point-to-multipoint interfaces
- Dynamic routing protocols configured on st0 interfaces

When there are multiple traffic selectors configured for a route-based VPN, clear traffic may enter a VPN tunnel without matching a traffic selector if the IKE gateway external interface is moved to another virtual router (VR). The software does not handle the multiple asynchronous interface events generated when an IKE gateway external interface is moved to another VR. As a workaround, first deactivate the IPsec VPN tunnel and commit the configuration without that tunnel before moving the IKE gateway external interface to another VR.

From Junos OS Release 21.1R1 onwards, you can configure multiple sets of local IP prefix, remote IP prefix, source port range, destination port range, and protocol for traffic selection. This means, multiple sets of IP address ranges, port ranges, and protocols can be part of same traffic selector as defined in RFC 7296. When you configure multiple traffic selectors, each traffic selector leads to a separate negotiation that results in the multiple IPsec tunnels. But, if you configure multiple terms under one traffic selector, this configuration results in single IPsec SA negotiation with multiple IP prefixes, ports, and protocols. See Traffic Selector.

Understanding Auto Route Insertion

Auto route insertion (ARI) automatically inserts a static route for the remote network and hosts protected by a remote tunnel endpoint. A route is created based on the remote IP address configured in the traffic-selector. In the case of traffic selectors, the configured remote address is inserted as a route in the routing instance associated with the st0 interface that is bound to the VPN.

Routing protocols and traffic selector configuration are mutually exclusive ways of steering traffic to a tunnel. ARI routes might conflict with routes that are populated through routing protocols. Therefore, you should not configure routing protocols on an st0 interface that is bound to a VPN on which traffic selectors are configured.

ARI is also known as reverse route insertion (RRI). ARI routes are inserted in the routing table as follows:

- If the `establish-tunnels immediately` option is configured at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level, ARI routes are added after Phase 1 and Phase 2 negotiations are complete. Because a route is not added until SAs are established, a failed negotiation does not result in traffic being routed to a st0 interface that is down. An alternate or backup tunnel is used instead.

- If the `establish-tunnels immediately` option is not configured at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level, ARI routes are added at configuration commit.

- An ARI route is not added if the configured or negotiated remote address in a traffic selector is 0.0.0.0/0 or 0::0.

The preference for the static ARI route is 5. This value is necessary to avoid conflict with similar routes that might be added by a routing protocol process. There is no configuration of the metric for the static ARI route.

The static ARI route cannot be leaked to other routing instances using the `rib-groups` configuration. Use the `import-policy` configuration to leak static ARI routes.

Understanding Traffic Selectors and Overlapping IP Addresses

This section discusses overlapping IP addresses in traffic selector configurations.

Overlapping IP Addresses in Different VPNs Bound to the Same st0 Interface

This scenario is not supported with traffic selectors. Traffic selectors cannot be configured on different VPNs that are bound to the same point-to-multipoint st0 interface, as shown in the following example:

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
}
vpn vpn-2 {
    bind-interface st0.1;
}
```

Overlapping IP Addresses in the Same VPN Bound to the Same st0 Interface

When overlapping IP addresses are configured for multiple traffic selectors in the same VPN, the first configured traffic selector that matches the packet determines the tunnel used for packet encryption.

In the following example, four traffic selectors (ts-1, ts-2, ts-3, and ts-4) are configured for the VPN (vpn-1), which is bound to the point-to-point st0.1 interface:

```
[edit]
user@host# show security ipsec vpn vpn-1
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.5.0/24;
        remote-ip 10.1.5.0/24;
    }
    traffic-selector ts-2 {
        local-ip 192.168.0.0/16;
        remote-ip 10.1.0.0/16;
    }
    traffic-selector ts-3 {
        local-ip 172.16.0.0/16;
        remote-ip 10.2.0.0/16;
    }
    traffic-selector ts-4 {
        local-ip 172.16.5.0/24;
        remote-ip 10.2.5.0/24;
    }
}
```

A packet with a source address 192.168.5.5 and a destination address 10.1.5.10 matches traffic selectors ts-1 and ts-2. However, traffic selector ts-1 is the first configured match and the tunnel associated with ts-1 is used for packet encryption.

A packet with a source address 172.16.5.5 and a destination address 10.2.5.10 matches the traffic selectors ts-3 and ts-4. However, traffic selector ts-3 is the first configured match and the tunnel associated with traffic selector ts-3 is used for packet encryption.

Overlapping IP Addresses in Different VPNs Bound to Different st0 Interfaces

When overlapping IP addresses are configured for multiple traffic selectors in different VPNs that are bound to different point-to-point st0 interfaces, an st0 interface is first selected by the longest prefix match for a given packet. Within the VPN that is bound to the selected st0 interface, the traffic selector is then selected based on the first configured match for the packet.

In the following example, a traffic selector is configured in each of two VPNs. The traffic selectors are configured with the same local subnetwork but different remote subnetworks.

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.1.0/24;
        remote-ip 10.1.1.0/24;
    }
}
vpn vpn-2 {
    bind-interface st0.2;
    traffic-selector ts-2 {
        local-ip 192.168.1.0/24;
        remote-ip 10.2.2.0/24;
    }
}
```

Different remote subnetworks are configured in each traffic selector, therefore two different routes are added to the routing table. Route lookup uses the st0 interface bound to the appropriate VPN.

In the following example, a traffic selector is configured in each of two VPNs. The traffic selectors are configured with different remote subnetworks. The same local subnetwork is configured for each traffic selector, but different netmask values are specified.

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.0.0/8;
        remote-ip 10.1.1.0/24;
    }
}
vpn vpn-2 {
    bind-interface st0.2;
    traffic-selector ts-2 {
        local-ip 192.168.0.0/16;
        remote-ip 10.2.2.0/24;
    }
}
```

A different remote subnetwork is configured in each traffic selector, therefore two different routes are added to the routing table. Route lookup uses the st0 interface bound to the appropriate VPN.

In the following example, traffic selectors are configured in each of two VPNs. The traffic selectors are configured with different local and remote subnetworks.

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.1.0/24;
        remote-ip 10.1.1.0/24;
    }
}
vpn vpn-2 {
    bind-interface st0.2;
    traffic-selector ts-2 {
        local-ip 172.16.1.0/24;
        remote-ip 10.2.2.0/24;
    }
}
```

In this case, the traffic selectors do not overlap. The remote subnetworks configured in the traffic selectors are different, therefore two different routes are added to the routing table. Route lookup uses the st0 interface bound to the appropriate VPN.

In the following example, a traffic selector is configured in each of two VPNs. The traffic selectors are configured with the same local subnetwork. The same remote subnetwork is configured for each traffic selector, but different netmask values are specified.

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.1.0/24;
        remote-ip 10.1.1.0/24;
    }
}
vpn vpn-2 {
    bind-interface st0.2;
    traffic-selector ts-2 {
        local-ip 192.168.1.0/24;
        remote-ip 10.1.0.0/16;
    }
}
```

Note that the `remote-ip` configured for ts-1 is 10.1.1.0/24 while the `remote-ip` configured for ts-2 is 10.1.0.0/16. For a packet destined to 10.1.1.1, route lookup selects the st0.1 interface as it has the longer prefix match. The packet is encrypted based on the tunnel corresponding to the st0.1 interface.

In some cases, valid packets can be dropped due to traffic selector traffic enforcement. In the following example, traffic selectors are configured in each of two VPNs. The traffic selectors are configured with different local subnetworks. The same remote subnetwork is configured for each traffic selector, but different netmask values are specified.

```
[edit]
user@host# show security ipsec
vpn vpn-1 {
    bind-interface st0.1;
    traffic-selector ts-1 {
        local-ip 192.168.1.0/24;
        remote-ip 10.1.1.0/24;
    }
}
vpn vpn-2 {
    bind-interface st0.2;
    traffic-selector ts-2 {
        local-ip 172.16.1.0/16;
        remote-ip 10.1.0.0/16;
    }
}
```

Two routes to 10.1.1.0 (10.1.1.0/24 via interface st0.1 and 10.1.0.0/16 via interface st0.2) are added to the routing table. A packet sent from source 172.16.1.1 to destination 10.1.1.1 matches the routing table entry for 10.1.1.0/24 via interface st0.1. However, the packet does not match the traffic specified by traffic selector ts-1 and is dropped.

If multiple traffic selectors are configured with the same remote subnetwork and netmask, equal cost routes are added to the routing table. This case is not supported with traffic selectors as the route chosen cannot be predicted.

SEE ALSO

Understanding VPN Tunnel Modes | **496**

Example: Configuring Traffic Selectors in a Route-Based VPN

IN THIS SECTION

This example shows how to configure traffic selectors for a route-based VPN.

Requirements

Before you begin,

- Read "Understanding Traffic Selectors in Route-Based VPNs" on page 524.

- Install the IKE package.

```
user@host> request system software add optional://junos-ike.tgz
```

Overview

IN THIS SECTION

This example configures traffic selectors to allow traffic to flow between subnetworks on SRX_A and subnetworks on SRX_B.

Table 61 on page 532 shows the traffic selectors for this example. Traffic selectors are configured under Phase 2 options.

Table 61: Traffic Selector Configurations

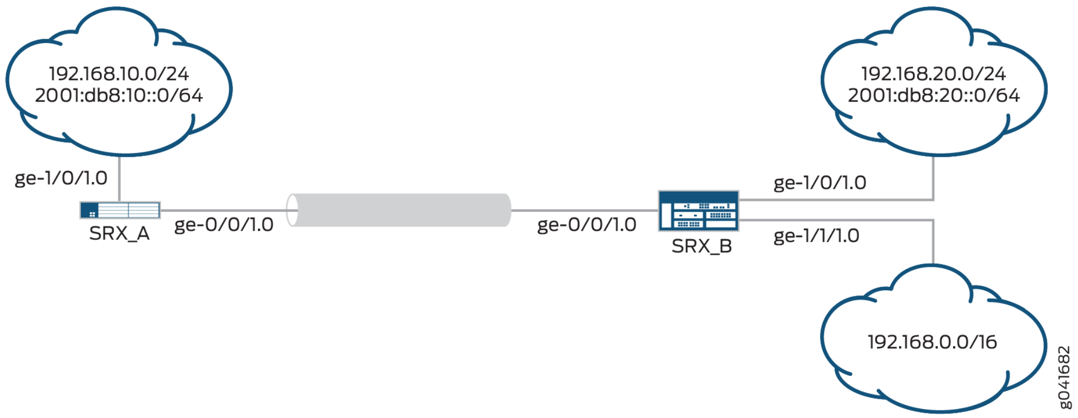
SRX_A			SRX_B		
Traffic Selector Name	Local IP	Remote IP	Traffic Selector Name	Local IP	Remote IP
TS1-ipv6	2001:db8:10::0/64	2001:db8:20::0/64	TS1-ipv6	2001:db8:20::0/64	2001:db8:10::0/64
TS2-ipv4	192.168.10.0/24	192.168.0.0/16	TS2-ipv4	192.168.0.0/16	192.168.10.0/24

Flow-based processing of IPv6 traffic must be enabled with the `mode flow-based` configuration option at the `[edit security forwarding-options family inet6]` hierarchy level.

Topology

In Figure 44 on page 533, an IPv6 VPN tunnel carries both IPv4 and IPv6 traffic between the SRX_A and SRX_B devices. That is, the tunnel operates in both IPv4-in-IPv6 and IPv6-in-IPv6 tunnel modes.

Figure 44: Traffic Selector Configuration Example



Configuration

IN THIS SECTION

Configuring SRX_A

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter commit from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:2000::1/64
set interfaces st0 unit 1 family inet
set interfaces st0 unit 1 family inet6
set interfaces ge-1/0/1 unit 0 family inet address 192.168.10.1/24
set interfaces ge-1/0/1 unit 0 family inet6 address 2001:db8:10::0/64
set security ike proposal PSK-DH14-AES256-SHA256 authentication- method pre-shared-keys
set security ike proposal PSK-DH14-AES256-SHA256 dh-group group14
set security ike proposal PSK-DH14-AES256-SHA256 authentication- algorithm sha-256
set security ike proposal PSK-DH14-AES256-SHA256 encryption-algorithm aes-256-cbc
set security ike policy site-2-site mode main
set security ike policy site-2-site proposals PSK-DH14-AES256-SHA256
set security ike policy site-2-site pre-shared-key ascii-text "$ABC123"
set security ike gateway SRX_A-to-SRX_B ike-policy site-2-site
set security ike gateway SRX_A-to-SRX_B address 192.168.20.2
set security ike gateway SRX_A-to-SRX_B external-interface ge-0/0/1.0
set security ike gateway SRX_A-to-SRX_B local-address 192.168.10.1
set security ipsec proposal ESP-AES256-SHA256 protocol esp
set security ipsec proposal ESP-AES256-SHA256 authentication- algorithm hmac-sha-256-128
set security ipsec proposal ESP-AES256-SHA256 encryption-algorithm aes-256-cbc
set security ipsec policy site-2-site perfect-forward-secrecy keys group14
set security ipsec policy site-2-site proposals ESP-AES256-SHA256
set security ipsec vpn SRX_A-to-SRX_B bind-interface st0.1
set security ipsec vpn SRX_A-to-SRX_B ike ipsec-policy site-2-site
set security ipsec vpn SRX_A-to-SRX_B ike gateway SRX_A-to-SRX_B
set security ipsec vpn SRX_A-to-SRX_B traffic-selector TS1-ipv6 term term1 local-ip
2001:db8:10::0/64 remote-ip 2001:db8:20::0/64
set security ipsec vpn SRX_A-to-SRX_B traffic-selector TS2-ipv4 term term2 local-ip
192.168.10.0/24 remote-ip 192.168.0.0/16
set security forwarding-options family inet6 mode flow-based
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-1/0/1.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone VPN interfaces st0.1
set security policies from-zone VPN to-zone trust policy 1 match source-address any
set security policies from-zone VPN to-zone trust policy 1 match destination-address any
set security policies from-zone VPN to-zone trust policy 1 match application any
set security policies from-zone VPN to-zone trust policy 1 then permit
set security policies from-zone trust to-zone VPN policy 1 match source-address any
set security policies from-zone trust to-zone VPN policy 1 match destination-address any
set security policies from-zone trust to-zone VPN policy 1 match application any
set security policies from-zone trust to-zone VPN policy 1 then permit
set security policies default-policy deny -all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure traffic selectors:

1. Configure the external interface.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:2000::1/64
```

2. Configure the secure tunnel interface.

```
[edit interfaces]
user@host# set st0 unit 1 family inet
user@host# set st0 unit 1 family inet6
```

3. Configure the internal interface.

```
[edit interfaces]
user@host# set ge-1/0/1 unit 0 family inet address 192.168.10.1/24
user@host# set ge-1/0/1 unit 0 family inet6 address 2001:db8:10::0/64
```

4. Configure Phase 1 options.

```
[edit security ike proposal PSK-DH14-AES256-SHA256]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy site-2-site]
user@host# set mode main
user@host# set proposals PSK-DH14-AES256-SHA256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security ike gateway SRX_A-to-SRX_B]
user@host# set ike-policy site-2-site
user@host# set address 192.168.20.2
user@host# set external-interface ge-0/0/1.0
user@host# set local-address 192.168.10.1
```

5. Configure Phase 2 options.

```
[edit security ipsec proposal ESP-AES256-SHA256]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy site-2-site]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ESP-AES256-SHA256
[edit security ipsec vpn SRX_A-to-SRX_B]
user@host# set bind-interface st0.1
user@host# set ike gateway SRX_A-to-SRX_B
user@host# set ike ipsec-policy site-2-site
user@host# set traffic-selector TS1-ipv6 term term1 local-ip 2001:db8:10::0/64 remote-ip
2001:db8:20::0/64
user@host# set traffic-selector TS2-ipv4 term term2 local-ip 192.168.10.0/24 remote-ip
192.168.0.0/16
```

6. Enable IPv6 flow-based forwarding.

```
[edit security forwarding-options]
user@host# set family inet6 mode flow-based
```

7. Configure security zones and the security policy.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic  system-services all
user@host# set host-inbound-traffic  protocols all
user@host# set interfaces ge-1/0/1.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic  system-services ike
user@host# set interfaces ge-0/0/1.0
[edit security zones security-zone VPN]
user@host# set interfaces st0.1
[edit security policies from-zone VPN to-zone trust ]
user@host# set policy 1 match source-address any
user@host# set policy 1 match destination-address any
user@host# set policy 1 match application any
user@host# set policy 1 then permit
[edit security policies from-zone trust to-zone VPN ]
```

```
user@host# set policy 1 match source-address any
user@host# set policy 1 match destination-address any
user@host# set policy 1 match application any
user@host# set policy 1 then permit
[edit security policies]
user@host# set default-policy deny-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike`, `show security ipsec`, `show security forwarding-options`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
    user@host# show interfaces
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:2000::1/64;
            }
        }
    }
    ge-1/0/1 {
        unit 0 {
            family inet {
                address 192.168.10.1/24;
            }
            family inet6 {
                address 10::1/64;
            }
        }
    }
    st0 {
        unit 1 {
            family inet;
            family inet6;
        }
    }
    [edit]
    user@host# show security ike
```

```
proposal PSK-DH14-AES256-SHA256 {
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
}
policy site-2-site {
    mode main;
    proposals PSK-DH14-AES256-SHA256;
        pre-shared-key ascii-text
    "$ABC123"; ## SECRET-DATA
}
gateway SRX_A-to-SRX_B {
    ike-policy site-2-site;
    address 192.168.20.2;
    external-interface ge-0/0/1.0;
    local-address 192.168.10.1;
}
[edit]
user@host# show security ipsec
proposal ESP-AES256-SHA256 {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-256-cbc;
}
policy site-2-site {
    perfect-forward-secrecy keys group14;
    proposals ESP-AES256-SHA256;
}
vpn SRX_A-to-SRX_B {
    bind-interface st0.1;
    ike {
        ipsec-policy site-2-site;
        gateway SRX_A-to-SRX_B;
    }
    traffic-selector TS1-ipv6 {
        local-ip 2001:db8:10::0/64;
        remote-ip 2001:db8:20::0/64;
    }
    traffic-selector TS2-ipv4 {
        local-ip 192.168.10.0/24;
        remote-ip 192.168.0.0/16;
    }
    }
    [edit]
    user@host# show security forwarding-options
    family {
        inet6 {
            mode flow-based;
        }
    }
    [edit]
    user@host# show security zones
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-1/0/1.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                ike;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
    security-zone VPN {
        interfaces {
            st0.1;
        }
    }
    [edit]
user@host# show security policies
    from-zone VPN to-zone trust {
        policy 1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone trust to-zone VPN {
        policy 1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring SRX_B

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:2000::2/64
set interfaces st0 unit 1 family inet
set interfaces st0 unit 1 family inet6
set interfaces ge-1/0/1 unit 0 family inet address 192.168.20.1/24
set interfaces ge-1/0/1 unit 0 family inet6 address 2001:db8:20::0/64
set interfaces ge-1/1/1 unit 0 family inet address 192.168.0.1/24
set security ike proposal PSK-DH14-AES256-SHA256 authentication-method pre-shared-keys
set security ike proposal PSK-DH14-AES256-SHA256 dh-group group14
set security ike proposal PSK-DH14-AES256-SHA256 authentication-algorithm sha-256
set security ike proposal PSK-DH14-AES256-SHA256 encryption-algorithm aes-256-cbc
set security ike policy site-2-site mode main
set security ike policy site-2-site proposals PSK-DH14-AES256-SHA256
set security ike policy site-2-site pre-shared-key ascii-text "$ABC123"
set security ike gateway SRX_B-to-SRX_A ike-policy site-2-site
set security ike gateway SRX_B-to-SRX_A address 192.168.10.1
set security ike gateway SRX_B-to-SRX_A external-interface ge-0/0/1.0
set security ike gateway SRX_B-to-SRX_A local-address 192.168.20.2
set security ipsec proposal ESP-AES256-SHA256 protocol esp
set security ipsec proposal ESP-AES256-SHA256 authentication-algorithm hmac-sha-256-128
set security ipsec proposal ESP-AES256-SHA256 encryption-algorithm aes-256-cbc
set security ipsec policy site-2-site perfect-forward-secrecy keys group14
set security ipsec policy site-2-site proposals ESP-AES256-SHA256
set security ipsec vpn SRX_B-to-SRX-A bind-interface st0.1
set security ipsec vpn SRX_B-to-SRX-A ike ipsec-policy site-2-site
set security ipsec vpn SRX_B-to-SRX-A ike gateway SRX_B-to-SRX_A
set security ipsec vpn SRX_B-to-SRX-A traffic-selector TS1-ipv6 local-ip 2001:db8:20::0/64
remote-ip 2001:db8:10::0/64
set security ipsec vpn SRX_B-to-SRX-A traffic-selector TS2-ipv4 local-ip 192.168.0.0/16 remote-
ip 192.168.10.0/24
set security forwarding-options family inet6 mode flow-based
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-1/0/1.0
set security zones security-zone trust interfaces ge-1/1/1.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone VPN interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/1.0
set security policies from-zone VPN to-zone trust policy 1 match source-address any
set security policies from-zone VPN to-zone trust policy 1 match destination-address any
set security policies from-zone VPN to-zone trust policy 1 match application any
set security policies from-zone VPN to-zone trust policy 1 then permit
set security policies from-zone trust to-zone VPN policy 1 match source-address any
set security policies from-zone trust to-zone VPN policy 1 match destination-address any
set security policies from-zone trust to-zone VPN policy 1 match application any
set security policies from-zone trust to-zone VPN policy 1 then permit
set security policies default-policy deny -all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure traffic selectors:

1. Configure the external interface.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:2000::2/64
```

2. Configure the secure tunnel interface.

```
[edit interfaces]
user@host# set st0 unit 1 family inet
user@host# set st0 unit 1 family inet6
```

3. Configure the internal interfaces.

```
[edit interfaces]
user@host# set ge-1/0/1 unit 0 family inet address 192.168.20.1/24
user@host# set ge-1/0/1 unit 0 family inet6 address 2001:db8:20::0/64
user@host# set ge-1/1/1 unit 0 family inet address 192.168.0.1/24
```

4. Configure Phase 1 options.

```
[edit security ike proposal PSK-DH14-AES256-SHA256]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy site-2-site]
user@host# set mode main
user@host# set proposals PSK-DH14-AES256-SHA256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security ike gateway SRX_B-to-SRX_A]
user@host# set ike-policy site-2-site
user@host# set address 192.168.10.1
user@host# set external-interface ge-0/0/1.0
user@host# set local-address 192.168.20.2
```

5. Configure Phase 2 options.

```
[edit security ipsec proposal ESP-AES256-SHA256]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy site-2-site]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ESP-AES256-SHA256
[edit security ipsec vpn SRX_B-to-SRX-A]
user@host# set bind-interface st0.1
user@host# set ike gateway SRX_B-to-SRX_A
user@host# set ike ipsec-policy site-2-site
user@host# set traffic-selector TS1-ipv6 local-ip 2001:db8:20::0/64 remote-ip
2001:db8:10::0/64
user@host# set traffic-selector TS2-ipv4 local-ip 192.168.0.0/16 remote-ip 192.168.10.0/24
```

6. Enable IPv6 flow-based forwarding.

```
[edit security forwarding-options]
user@host# set family inet6 mode flow-based
```

7. Configure security zones and the security policy.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic  system-services all
user@host# set host-inbound-traffic  protocols all
user@host# set interfaces ge-1/0/1.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic  system-services ike
user@host# set interfaces ge-0/0/1.0
[edit security zones security-zone VPN]
user@host# set interfaces st0.1
[edit security policies from-zone VPN to-zone trust ]
user@host# set policy 1 match source-address any
user@host# set policy 1 match destination-address any
user@host# set policy 1 match application any
user@host# set policy 1 then permit
[edit security policies from-zone trust to-zone VPN ]
user@host# set policy 1 match source-address any
```

```
user@host# set policy 1 match destination-address any
user@host# set policy 1 match application any
user@host# set policy 1 then permit
[edit security policies]
user@host# set default-policy deny-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike`, `show security ipsec`, `show security forwarding-options`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
    user@host# show interfaces
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:2000::2/64;
            }
        }
    }
    ge-1/0/1 {
        unit 0 {
            family inet {
                address 192.168.20.1/24;
            }
            family inet6 {
                address 2001:db8:20::0/64;
            }
        }
    }
    ge-1/1/1 {
        unit 0 {
            family inet {
                address 192.168.0.1/24;
            }
        }
    }
    st0 {
        unit 1 {
```

```
        family inet;
        family inet6;
    }
}
[edit]
user@host# show security ike
proposal PSK-DH14-AES256-SHA256 {
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
}
policy site-2-site {
    mode main;
    proposals PSK-DH14-AES256-SHA256;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway SRX_B-to-SRX_A {
    ike-policy site-2-site;
    address 192.168.10.1;
    external-interface ge-0/0/1.0;
    local-address 192.168.20.2;
}
[edit]
user@host# show security ipsec
proposal ESP-AES256-SHA256 {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-256-cbc;
}
policy site-2-site {
    perfect-forward-secrecy keys group14;
    proposals ESP-AES256-SHA256;
}
vpn SRX_B-to-SRX-A {
    bind-interface st0.1;
    ike {
        ipsec-policy site-2-site;
        gateway SRX_B-to-SRX_A;
    }
    traffic-selector TS1-ipv6 {
        local-ip 2001:db8:20::0/64;
        remote-ip 2001:db8:10::0/64;
    }
    traffic-selector TS2-ipv4 {
        local-ip 192.168.0.0/16;
        remote-ip 192.168.10.0/24;
    }
}
[edit]
user@host# show security forwarding-options
family {
    inet6 {
        mode flow-based;
    }
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-1/0/1.0;
        ge-1/1/1.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
security-zone VPN {
    interfaces {
        st0.1;
    }
    }
    [edit]
user@host# show security policies
    from-zone VPN to-zone trust {
        policy 1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone trust to-zone VPN {
        policy 1 {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

The sample outputs shown are on SRX-A.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the `show security ipsec security-associations` command.

```
user@host> show security ipsec security-associations
  Total active tunnels: 3
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <268173313 ESP:3des/ sha-256 3d75aeff 2984/ unlim -  root 500   2001:db8:2000::2
  >268173313 ESP:3des/ sha-256 a468fece 2984/ unlim -  root 500   2001:db8:2000::2
  <268173316 ESP:3des/ sha-256 417f3cea 3594/ unlim -  root 500   2001:db8:2000::2
  >268173316 ESP:3des/ sha-256 a4344027 3594/ unlim -  root 500   2001:db8:2000::2
```

From operational mode, enter the `show security ipsec security-associations detail` command.

```
user@host> show security ipsec security-associations detail
  ID: 268173313 Virtual-system: root, VPN Name: SRX_A-to-SRX_B
  Local Gateway: 192.168.10.1, Remote Gateway: 2192.168.20.2
  Traffic Selector Name: TS1-ipv6
  Local Identity: ipv6(2001:db8:10::-2001:db8:10::ffff:ffff:ffff:ffff)
  Remote Identity: ipv6(2001:db8:20::-2001:db8:20::ffff:ffff:ffff:ffff)
  Version: IKEv1
    DF-bit: clear
    Bind-interface: st0.1

  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: c608b29
  Tunnel Down Reason: SA not initiated
    Direction: inbound, SPI: 3d75aeff, AUX-SPI: 0
                            , VPN Monitoring: -
    Hard lifetime: Expires in 2976 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2354 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256-128, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64

    Direction: outbound, SPI: a468fece, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2976 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2354 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256-128, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64

  ID: 268173316 Virtual-system: root, VPN Name: SRX_A-to-SRX_B
  Local Gateway: 192.168.10.1, Remote Gateway: 192.168.20.2
  Traffic Selector Name: TS2-ipv4
  Local Identity: ipv4(192.168.10.0-192.168.10.255)
  Remote Identity: ipv4(192.168.20.0-192.168.20.255)
  Version: IKEv1
    DF-bit: clear
    Bind-interface: st0.1

  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: c608b29
  Tunnel Down Reason: SA not initiated
    Direction: inbound, SPI: 417f3cea, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3586 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2948 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256-128, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64

    Direction: outbound, SPI: a4344027, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3586 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2948 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256-128, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the peer devices.

Verifying Traffic Selectors

Purpose

Verify negotiated traffic selectors on the secure tunnel interface.

Action

From operational mode, enter the `show security ipsec traffic-selector st0.1` command.

```
user@host> show security ipsec traffic-selector st0.1
Source IP                                         Destination
IP                                                          Interface   Tunnel-id         IKE-ID
2001:db8:10::-2001:db8:10::ffff:ffff:ffff:ffff
2001:db8:20::-2001:db8:20::ffff:ffff:ffff:ffff      st0.1       268173313      2001:db8:2000::1
192.168.10.0-192.168.10.255
192.168.0.0-192.168.255.255                              st0.1       268173316
2001:db8:2000::1
192.168.10.0-192.168.10.255
192.168.20.0-192.168.20.255                              st0.1       268173317
2001:db8:2000::1
```

Verifying Routes

Purpose

Verify active routes

Action

From operational mode, enter the `show route` command.

```
user@host> show route
inet.0: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both

192.168.0.0/16        *[ARI-TS/5] 00:00:32
                    > via st0.1
2001:db8:20::0/64      *[ARI-TS/5] 00:00:34
                    > via st0.1
```

Meaning

The `show route` command lists active entries in the routing tables. Routes to the remote IP address configured in each traffic selector should be present with the correct st0 interface.

SEE ALSO

Understanding VPN Tunnel Modes | **496**

Release History Table

Release	Description
15.1X49-D140	Starting with Junos OS Release 15.1X49-D140, on all SRX Series Firewalls and vSRX Virtual Firewall instances, when you configure the traffic-selector with a remote address of 0::0 (IPv6), the following **"error: configuration check-out failed"** message is displayed when performing the commit and the configuration checkout fails.
15.1X49-D100	Starting with Junos OS Release 15.1X49-D100, traffic selectors can be configured with IKEv2 site-to-site VPNs.
12.1X46-D10	Starting with Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, traffic selectors can be configured with IKEv1 site-to-site VPNs.

RELATED DOCUMENTATION

Route-Based IPsec VPNs | **394**

8

CHAPTER

Class-Of-Service Based VPN

CoS-Based IPsec VPNs

IN THIS SECTION

You can configure Junos class-of-service (CoS) features to provide multiple classes of service for VPNs. On the device, you can configure multiple forwarding classes for transmitting packets, define which packets are placed into each output queue, schedule the transmission service level for each queue, and manage congestion.

Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs

IN THIS SECTION

Class of service (CoS) forwarding classes (FCs) configured on the SRX Series Firewall can be mapped to IPsec security associations (SAs). Packets for each FC are mapped to a different IPsec SA, thus providing for CoS treatment on the local device and on intermediate routers.

Benefits of CoS-Based IPsec VPNs with Multiple IPsec SAs

- Helps you ensure different data streams, with each tunnel using a separate set of security associations.
- Helps you to facilitate the IPsec VPN deployments where differentiated traffic is required, such as voice-over-IP.

Overview

This feature is proprietary to Juniper Networks and works with supported SRX platforms and Junos OS releases. The VPN peer device must be an SRX Series Firewall or vSRX Virtual Firewall instance that supports this feature or any other product that support the same functionality in the same way as SRX Series Firewall.

Mapping FCs to IPsec SAs

Up to 8 forwarding classes (FC) can be configured for a VPN with the `multi-sa forwarding-classes` at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level. The number of IPsec SAs negotiated with a peer gateway is based on the number of FCs configured for the VPN. The mapping of FCs to IPsec SAs applies to all traffic selectors configured for the VPN.

All IPsec SAs created for the FCs of a specific VPN are represented by the same tunnel ID. Tunnel-related events consider the state and statistics of all IPsec SAs. All IPsec SAs related to a tunnel are anchored to the same SPU or the same thread ID on SRX Series Firewalls or vSRX Virtual Firewall instances.

IPsec SA Negotiation

When multiple FCs are configured for a VPN, a unique IPsec SA is negotiated with the peer for each FC. In addition, a default IPsec SA is negotiated to send packets that do not match a configured FC. The default IPsec is negotiated even If the VPN peer device is not configured for FCs or does not support FC to IPsec SA mapping. The default IPsec SA is the first IPsec SA to be negotiated and the last SA to be torn down.

Depending on the number of FCs configured. When IPsec SAs are in the process of negotiating, packets may arrive with an FC for which an IPsec SA has yet to be negotiated. Until an IPsec SA for a given FC is negotiated, the traffic is sent to the default IPsec SA. A packet with an FC that does not match any of the installed IPsec SAs is sent on the default IPsec SA.

Mapping of FCs to IPsec SAs is done on the local VPN gateway. The local and peer gateways may have FCs configured in a different order. Each peer gateway maps FCs in the order in which IPsec SA negotiations are completed. Thus, the local and peer gateways might have different FC to IPsec SA mappings. A gateway stops negotiating new IPsec SAs once the configured number of FCs is reached. A peer gateway may initiate more IPsec SAs than the number of FCs configured on the local gateway. In this case, the local gateway accepts the additional IPsec SA requests—up to 18 IPsec SAs. The local gateway uses the other IPsec SAs only for decrypting incoming IPsec traffic. If a packet is received with an FC that does not match any configured FC, the packet is sent on the default FC IPsec SA.

If a delete notification is received for the default IPsec SA from the peer device, only the default IPsec SA is deleted and the default IPsec SA is negotiated newly. During this time, traffic which might go on default IPsec SA is be dropped. The VPN tunnel is brought down only if the default IPsec SA is the last SA.

If the `establish-tunnels immediately` option is configured and committed for the VPN, the SRX Series Firewall negotiates IPsec SA without waiting for traffic to arrive. If negotiations do not complete for an IPsec SA for a configured FC, negotiations are retried every 60 seconds.

If the `establish-tunnels on-traffic` option is configured for the VPN, the SRX Series Firewall negotiates IPsec SAs when the first data packet arrives; the FC for the first packet does not matter. With either option, the default IPsec SA is negotiated first, then each IPsec SA is negotiated one by one in the order in which the FCs are configured on the device.

Rekey

When using Multi SAs with Differentiated Services Code Point (DSCP) traffic steering with traffic selectors, the following behavior occurs during rekey. When the traffic selectors performs rekeying, if one or more of the traffic selectors are unable to rekey for any reason, the specific SA is brought down when the lifetime expire. In this case, traffic that use to match the specific SA is sent through the default traffic selector instead.

Adding or Deleting FCs from a VPN

When FCs are added or deleted from a VPN, the IKE and IPsec SAs for the VPN are brought up or down and restarts the negotiations. The `clear security ipsec security-associations` command clears all IPsec SAs.

Dead Peer Detection (DPD)

When DPD is configured with this feature, the `optimized` mode sends probes only when there is outgoing traffic and no incoming traffic on any of the IPsec SA. While the `probe-idle` mode sends probes only when there is no outgoing and no incoming traffic on any of the IPsec SAs. VPN monitoring is not supported with DPD feature.

Commands

The `show security ipsec sa details index` *`tunnel-id`* command displays all IPsec SA details including the FC name. The `show security ipsec stats index` *`tunnel-id`* command displays statistics for each FC.

Supported VPN Features

The following VPN features are supported with CoS-based IPsec VPNs:

- Route-based site-to-site VPNs. Policy-based VPNs are not supported.
- AutoVPN.
- Traffic selectors.
- Auto Discovery VPNs (ADVPNs).
- IKEv2. IKEv1 is not supported.
- Dead peer detection (DPD). VPN monitoring is not supported.

SEE ALSO

Understanding Traffic Selectors and CoS-Based IPsec VPNs

Example: Configuring CoS-Based IPsec VPNs

Forwarding Classes Overview

Understanding Traffic Selectors and CoS-Based IPsec VPNs

A traffic selector is an agreement between IKE peers to permit traffic through a VPN tunnel if the traffic matches a specified pair of local and remote addresses. Only traffic that conforms to a traffic selector is permitted through the associated security association (SA).

The CoS-based IPsec VPN feature supports the following scenarios

- One or multiple traffic selectors in a route-based site-to-site VPN with the same FCs.
- Multiple traffic selectors, with different FCs for each traffic selector. This scenario requires separate VPN configurations.

This topic describes the VPN configurations and the IPsec SA that are negotiated for each scenario.

In the following scenarios, three FCs are configured on the SRX Series Firewall:

```
forwarding-classes {
    queue 7 voip-data;
    queue 6 web-data;
    queue 5 control-data;
                    }
```

In the first scenario, VPN vpn1 is configured with a single traffic selector ts1 and the three FCs:

```
ipsec {
    vpn vpn1 {
       ts1 {
           local-ip 3.3.3.0/24;
           remote-ip 4.4.4.0/24;
              }

    multi-sa {
        forwarding-class web-data;
           forwarding-class voip-data
           forwarding-class control-data;
          }
       }
   }
```

In the configuration above, four IPsec SAs are negotiated for traffic selector ts1—one for the default IPsec SA and three for the IPsec SAs that are mapped to FCs.

In the second scenario, VPN vpn1 is configured with two traffic selectors ts1 and ts2 and the three FCs:

```
ipsec {
    vpn vpn1 {
       ts1 {
           local-ip 3.3.3.0/24;
           remote-ip 4.4.4.0/24;
              }
  ts2 {
           local-ip 6.6.6.0/24;
           remote-ip 7.7.7.0/24;
         }
    multi-sa {
        forwarding-class web-data;
           forwarding-class voip-data
           forwarding-class control-data;
          }
       }
   }
```

In the configuration above, four IPsec SAs are negotiated for traffic selector ts1 and four IPsec SAs are negotiated for traffic selector ts2. For each traffic selector, there is one IPsec SA negotiated for the default IPsec SA and three IPsec SAs negotiated for the IPsec SAs that are mapped to FCs.

In the third scenario, traffic selectors ts1 and ts2 support different sets of FCs. The traffic selectors need to be configured for different VPNs:

```
ipsec {
    vpn vpn1 {
        bind-interface st0.0;
        ts1 {
            local-ip 3.3.3.0/24;
            remote-ip 4.4.4.0/24;
            }

        multi-sa {
         forwarding-class web-data;
             forwarding-class voip-data;
           forwarding-class control-data;
                 }
     vpn vpn2 {
         bind-interface st0.0;
         ts2 {
             local-ip 6.6.6.0/24;
             remote-ip 7.7.7.0/24;
             }
       multi-sa {
        forwarding-class web-data;
            forwarding-class voip-data;
             }
         }
```

In the configuration above, four IPsec SAs are negotiated for traffic selector ts1 in VPN vpn1—one for the default IPsec SA and three for the IPsec SAs that are mapped to FCs.

SEE ALSO

Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs

Example: Configuring CoS-Based IPsec VPNs

Example: Configuring CoS-Based IPsec VPNs

IN THIS SECTION

This example shows how to configure a CoS-based IPsec VPNs with multiple IPsec SAs to allow packets mapping for each forwarding class to a different IPsec SA, thus providing for CoS treatment on the local device and on intermediate routers.

This feature is proprietary to Juniper Networks and only works with supported SRX platforms and Junos OS releases. The VPN peer device must be an SRX Series Firewall or vSRX Virtual Firewall instance that supports this feature.

Requirements

This example uses the following hardware:

- Any SRX Series Firewall

Before you begin:

- Understand how Class of service (CoS) forwarding classes (FCs) configured on the SRX Series Firewall can be mapped to IPsec security associations (SAs). See Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs.

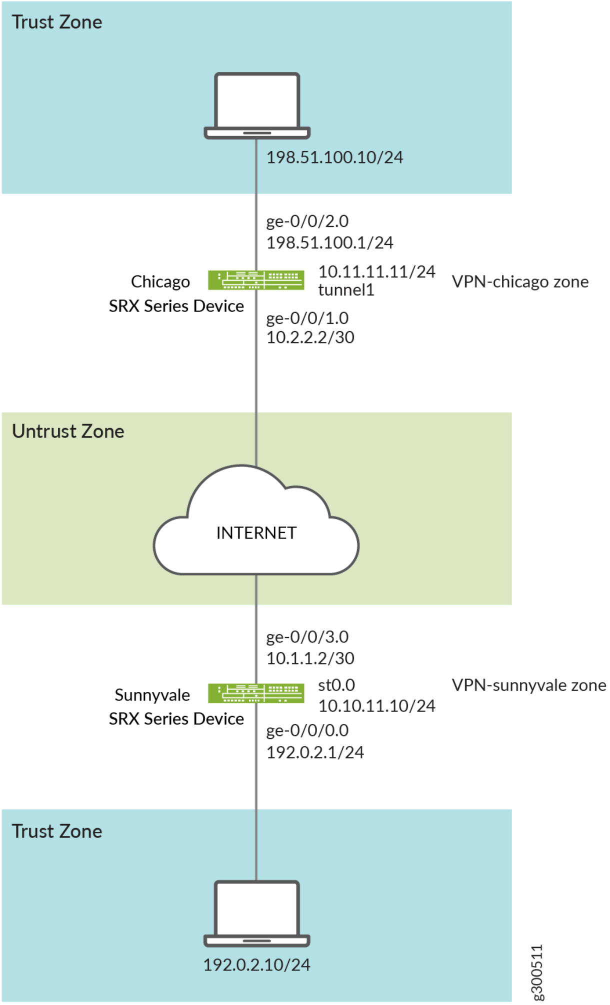
- Understand Traffic Selectors and CoS-Based IPsec VPNs. See Understanding Traffic Selectors and CoS-Based IPsec VPNs.

Overview

In this example, you configure an IPsec route-based VPN for a branch office in Chicago, because you do not need to conserve tunnel resources or configure many security policies to filter traffic through the tunnel. Users in the Chicago office will use the VPN to connect to their corporate headquarters in Sunnyvale.

Figure 45 on page 560 shows an example of an IPsec route-based VPN topology. In this topology, one SRX Series Firewall is located in Sunnyvale, and one SRX Series Firewall is located in Chicago.

Figure 45: IPsec Route-Based VPN Topology



In this example, you configure interfaces, an IPv4 default route and security zones. Then you configure IKE, IPsec, a security policy, and CoS parameters. See Table 62 on page 561 through Table 65 on page 563.

Table 62: Interface, Static Route, and Security Zone Information

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	192.0.2.1/24
	ge-0/0/3.0	10.1.1.2/30
	st0.0 (tunnel interface)	10.10.11.10/24
Static routes	0.0.0.0/0 (default route)	The next hop is st0.0.
Security zones	trust	• All system services are allowed. • The ge-0/0/0.0 interface is bound to this zone.
	untrust	• All system services are allowed. • The ge-0/0/3.0 interface is bound to this zone.
	vpn	The st0.0 interface is bound to this zone.

Table 63: IKE Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ike-proposal	• Authentication method: rsa-signatures • Diffie-Hellman group: group14 • Authentication algorithm: sha-256 • Encryption algorithm: aes-256-cbc

Table 63: IKE Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Policy	ike-policy	• Mode: main • Proposal reference: ike-proposal • IKE policy authentication method: rsa-signatures
Gateway	gw-sunnyvale	• IKE policy reference: ike-policy • External interface: ge-0/0/3.0 • Gateway address: 10.2.2.2

Table 64: IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-sha-256 • Encryption algorithm: aes-256-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop
VPN	ipsec_vpn1	• IKE gateway reference: gw-chicago • IPsec policy reference: ipsec_pol

Table 65: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
The security policy permits traffic from the trust zone to the vpn zone.	vpn	• Match criteria: • source-address sunnyvale • destination-address chicago • application any • Action: permit
The security policy permits traffic from the vpn zone to the trust zone.	vpn	• Match criteria: • source-address chicago • destination-address sunnyvale • application any • Action: permit

Configuration

IN THIS SECTION

Configuring Basic Network and Security Zone Information

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.1/24
set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
set interfaces st0 unit 0 family inet address 10.10.11.10/24
set routing-options static route 0.0.0.0/0 next-hop st0.0
set security zones security-zone untrust interfaces ge-0/0/3.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone vpn-chicago interfaces st0.0
set security zones security-zone vpn-chicago host-inbound-traffic protocols all
set security zones security-zone vpn-chicago host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone untrust host-inbound-traffic protocols all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure interface, static route, and security zone information:

1. Configure Ethernet interface information.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.1/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 10.1.1.2/30
user@host# set interfaces st0 unit 0 family inet address 10.10.11.10/24
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop st0.0
```

3. Configure the untrust security zone.

```
[edit ]
user@host# edit security zones security-zone untrust
```

4. Specify allowed system services for the untrust security zone.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic protocols all
```

5. Assign an interface to the security zone.

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/3.0
```

6. Specify allowed system services for the security zone.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services ike
```

7. Configure the trust security zone.

```
[edit]
user@host# edit security zones security-zone trust
```

8. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/0.0
```

9. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
```

10. Configure the vpn security zone.

```
[edit]
user@host# edit security zones security-zone vpn
```

11. Assign an interface to the security zone.

```
[edit security zones security-zone vpn-chicago]
user@host# set interfaces st0.0
user@host# set host-inbound-traffic protocols all
user@host# set host-inbound-traffic system-services all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security zones` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 10.1.1.2/30;
        }
    }
}
st0 {
    unit 0 {
        family inet {
        address 10.10.11.10/24;
        }
    }
}
```

```
[edit]
user@host# show routing-options
static {
    route 0.0.0.0/0 next-hop st0.0;
}
```

```
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone vpn-chicago {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.0;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring CoS

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set class-of-service classifiers dscp ba-classifier import default
set class-of-service classifiers dscp ba-classifier forwarding-class best-effort loss-priority
high code-points 000000
set class-of-service classifiers dscp ba-classifier forwarding-class ef-class loss-priority high
code-points 000001
set class-of-service classifiers dscp ba-classifier forwarding-class af-class loss-priority high
code-points 001010
set class-of-service classifiers dscp ba-classifier forwarding-class network-control loss-
priority high code-points 000011
set class-of-service classifiers dscp ba-classifier forwarding-class res-class loss-priority
high code-points 000100
set class-of-service classifiers dscp ba-classifier forwarding-class web-data loss-priority high
code-points 000101
set class-of-service classifiers dscp ba-classifier forwarding-class control-data loss-priority
```

```
high code-points 000111
set class-of-service classifiers dscp ba-classifier forwarding-class voip-data loss-priority
high code-points 000110
set class-of-service forwarding-classes queue 7 voip-data
set class-of-service forwarding-classes queue 6 control-data
set class-of-service forwarding-classes queue 5 web-data
set class-of-service forwarding-classes queue 4 res-class
set class-of-service forwarding-classes queue 2 af-class
set class-of-service forwarding-classes queue 1 ef-class
set class-of-service forwarding-classes queue 0 best-effort
set class-of-service forwarding-classes queue 3 network-control
set class-of-service interfaces ge-0/0/3 unit 0 classifiers dscp ba-classifier
set class-of-service interfaces ge-0/0/3 unit 0 scheduler-map sched_1
set class-of-service scheduler-maps sched_1 forwarding-class voip-data scheduler Q7
set class-of-service scheduler-maps sched_1 forwarding-class control-data scheduler Q6
set class-of-service scheduler-maps sched_1 forwarding-class web-data scheduler Q5
set class-of-service scheduler-maps sched_1 forwarding-class res-class scheduler Q4
set class-of-service scheduler-maps sched_1 forwarding-class af-class scheduler Q2
set class-of-service scheduler-maps sched_1 forwarding-class ef-class scheduler Q1
set class-of-service scheduler-maps sched_1 forwarding-class best-effort scheduler Q0
set class-of-service scheduler-maps sched_1 forwarding-class network-control scheduler Q3
set class-of-service schedulers Q7 transmit-rate percent 5
set class-of-service schedulers Q7 priority strict-high
set class-of-service schedulers Q6 transmit-rate percent 25
set class-of-service schedulers Q6 priority high
set class-of-service schedulers Q5 transmit-rate remainder
set class-of-service schedulers Q5 priority high
set class-of-service schedulers Q4 transmit-rate percent 25
set class-of-service schedulers Q4 priority medium-high
set class-of-service schedulers Q3 transmit-rate remainder
set class-of-service schedulers Q3 priority medium-high
set class-of-service schedulers Q2 transmit-rate percent 10
set class-of-service schedulers Q2 priority medium-low
set class-of-service schedulers Q1 transmit-rate percent 10
set class-of-service schedulers Q1 priority medium-low
set class-of-service schedulers Q0 transmit-rate remainder
set class-of-service schedulers Q0 priority low
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure CoS:

1. Configure behavior aggregate classifiers for DiffServ CoS.

```
[edit class-of-service]
user@host# edit classifiers dscp ba-classifier
user@host# set import default
```

2. Configure a best-effort forwarding class classifier.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class best-effort loss-priority high code-points 000000
```

3. Define the DSCP value to be assigned to the forwarding class.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class ef-class loss-priority high code-points 000001
user@host# set forwarding-class af-class loss-priority high code-points 001010
user@host# set forwarding-class network-control loss-priority high code-points 000011
user@host# set forwarding-class res-class loss-priority high code-points 000100
user@host# set forwarding-class web-data loss-priority high code-points 000101
user@host# set forwarding-class control-data loss-priority high code-points 000111
user@host# set forwarding-class voip-data loss-priority high code-points 000110
```

4. Define eight forwarding classes (queue names) for the eight queues.

```
[edit class-of-service forwarding-classes]
user@host# set queue 7 voip-data
user@host# set queue 6 control-data
user@host# set queue 5 web-data
user@host# set queue 4 res-class
user@host# set queue 2 af-class
user@host# set queue 1 ef-class
user@host# set queue 0 best-effort
user@host# set queue 3 network-control
```

5. Configure classifiers on the ingress (ge) interfaces.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/3 unit 0 classifiers dscp ba-classifier
```

6. Apply the scheduler map to the ge interface.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/3 unit 0 scheduler-map sched_1
```

7. Configure the scheduler map to associate schedulers with defined forwarding classes.

```
[edit class-of-service]
user@host# set scheduler-maps sched_1 forwarding-class voip-data scheduler Q7
user@host# set scheduler-maps sched_1 forwarding-class control-data scheduler Q6
user@host# set scheduler-maps sched_1 forwarding-class web-data scheduler Q5
user@host# set scheduler-maps sched_1 forwarding-class res-class scheduler Q4
user@host# set scheduler-maps sched_1 forwarding-class af-class scheduler Q2
user@host# set scheduler-maps sched_1 forwarding-class ef-class scheduler Q1
user@host# set scheduler-maps sched_1 forwarding-class best-effort scheduler Q0
user@host# set scheduler-maps sched_1 forwarding-class network-control scheduler Q3
```

8. Define the schedulers with priority and transmit rates.

```
[edit set class-of-service]
user@host# set schedulers Q7 transmit-rate percent 5
user@host# set schedulers Q7 priority strict-high
user@host# set schedulers Q6 transmit-rate percent 25
user@host# set schedulers Q6 priority high
user@host# set schedulers Q5 transmit-rate remainder
user@host# set schedulers Q5 priority high
user@host# set schedulers Q4 transmit-rate percent 25
user@host# set schedulers Q4 priority medium-high
user@host# set schedulers Q3 transmit-rate remainder
user@host# set schedulers Q3 priority medium-high
user@host# set schedulers Q2 transmit-rate percent 10
user@host# set schedulers Q2 priority medium-low
user@host# set schedulers Q1 transmit-rate percent 10
user@host# set schedulers Q1 priority medium-low
```

```
user@host# set schedulers Q0 transmit-rate remainder
user@host# set schedulers Q0 priority low
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service
classifiers {
    dscp ba-classifier {
        import default;
        forwarding-class best-effort {
            loss-priority high code-points 000000;
        }
        forwarding-class ef-class {
            loss-priority high code-points 000001;
        }
        forwarding-class af-class {
            loss-priority high code-points 001010;
        }
        forwarding-class network-control {
            loss-priority high code-points 000011;
        }
        forwarding-class res-class {
            loss-priority high code-points 000100;
        }
        forwarding-class web-data {
            loss-priority high code-points 000101;
        }
        forwarding-class control-data {
            loss-priority high code-points 000111;
        }
        forwarding-class voip-data {
            loss-priority high code-points 000110;
        }
    }
}
forwarding-classes {
    queue 7 voip-data;
    queue 6 control-data;
    queue 5 web-data;
    queue 4 res-class;
    queue 2 af-class;
    queue 1 ef-class;
    queue 0 best-effort;
    queue 3 network-control;
}
interfaces {
    ge-0/0/3 {
        unit 0 {
            classifiers {
                dscp ba-classifier;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            scheduler-map sched_1;
        }
    }
}
scheduler-maps {
    sched_1 {
        forwarding-class voip-data scheduler Q7;
        forwarding-class control-data scheduler Q6;
        forwarding-class web-data scheduler Q5;
        forwarding-class res-class scheduler Q4;
        forwarding-class af-class scheduler Q2;
        forwarding-class ef-class scheduler Q1;
        forwarding-class best-effort scheduler Q0;
        forwarding-class network-control scheduler Q3;
    }
}
schedulers {
    Q7 {
        transmit-rate percent 5;
        priority strict-high;
    }
    Q6 {
        transmit-rate percent 25;
        priority high;
    }
    Q5 {
        transmit-rate {
            remainder;
        }
        priority high;
    }
    Q4 {
        transmit-rate percent 25;
        priority medium-high;
    }
    Q3 {
        transmit-rate {
            remainder;
        }
        priority medium-high;
    }
    Q2 {
        transmit-rate percent 10;
        priority medium-low;
    }
    Q1 {
        transmit-rate percent 10;
        priority medium-low;
    }
    Q0 {
        transmit-rate {
            remainder;
        }
        priority low;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike-proposal authentication-method pre-shared-keys
set security ike proposal ike-proposal dh-group group14
set security ike proposal ike-proposal authentication-algorithm sha-256
set security ike proposal ike-proposal encryption-algorithm aes-256-cbc
set security ike policy ike-policy mode main
set security ike policy ike-policy proposals ike-proposal
set security ike policy ike-policy pre-shared-key ascii-text $ABC123
set security ike gateway gw-sunnyvale external-interface ge-0/0/3.0
set security ike gateway gw-sunnyvale ike policy ike-policy
set security ike gateway gw-sunnyvale address 10.2.2.2
set security ike gateway gw-sunnyvale version v2-only
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE:

1. Create the IKE proposal.

   ```
   [edit security ike]
   user@host# set proposal ike-proposal
   ```

2. Define the IKE proposal authentication method.

   ```
   [edit security ike proposal ike-proposal]
   user@host# set authentication-method pre-shared-keys
   ```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike-proposal]
user@host# set dh-group group14
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-algorithm sha-256
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike-proposal]
user@host# set encryption-algorithm aes-256-cbc
```

6. Create an IKE policy.

```
[edit security ike]
user@host# set policy ike-policy
```

7. Set the IKE policy mode.

```
[edit security ike policy ike-policy]
user@host# set mode main
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ike-policy]
user@host# set proposals ike-proposal
```

9. Define the IKE policy authentication method.

```
[edit security ike policy ike-policy]
user@host# set pre-shared-key ascii-text $ABC123
```

10. Create an IKE gateway and define its external interface.

    ```
    [edit security ike]
    user@host# set gateway gw-sunnyvale external-interface ge-0/0/3.0
    ```

11. Define the IKE policy reference.

    ```
    [edit security ike gateway gw-sunnyvale]
    user@host# set ike policy ike-policy
    ```

12. Define the IKE gateway address.

    ```
    [edit security ike gateway gw-sunnyvale]
    user@host# set address 10.2.2.2
    ```

13. Define the IKE gateway version.

    ```
    [edit security ike gateway gw-sunnyvale]
    user@host# set version v2-only
    ```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
}
policy ike-policy {
    mode main;
    proposals ike-proposal;
    pre-shared-key ascii-text "$ABC123";
```

```
}
gateway gw-sunnyvale {
    ike policy ike-policy;
    address 10.2.2.2;
    external-interface ge-0/0/3.0;
    version v2-only;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec traceoptions flag all
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha-256
set security ipsec proposal ipsec_prop encryption-algorithm aes256-cbc
set security ipsec proposal ipsec_prop lifetime-seconds 3600
set security ipsec policy ipsec_pol proposals ipsec_prop
set security ipsec vpn ipsec_vpn1 bind-interface st0.0
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class ef-class
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class af-class
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class res-class
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class web-data
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class control-data
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class voip-data
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class network-control
set security ipsec vpn ipsec_vpn1 multi-sa forwarding-class best-effort
set security ipsec vpn ipsec_vpn1 ike gateway gw_sunnyvale
set security ipsec vpn ipsec_vpn1 ike ipsec-policy ipsec_pol
set security ipsec vpn ipsec_vpn1 establish-tunnels immediately
set security ipsec vpn ipsec_vpn1 traffic-selector ipsec_vpn1_TS1 local-ip 203.0.113.2/25
set security ipsec vpn ipsec_vpn1 traffic-selector ipsec_vpn1_TS1 remote-ip 192.0.2.30/24
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec:

1. Enable IPsec trace options.

```
[edit]
user@host# set security ipsec traceoptions flag all
```

2. Create an IPsec proposal.

```
[edit]
user@host# set security ipsec proposal ipsec_prop
```

3. Specify the IPsec proposal protocol.

```
[edit security ipsec proposal ipsec_prop]
user@host# set protocol esp
```

4. Specify the IPsec proposal authentication algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set authentication-algorithm hmac-sha-256
```

5. Specify the IPsec proposal encryption algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set encryption-algorithm aes256-cbc
```

6. Specify the lifetime (in seconds) of an IPsec security association (SA).

```
[set security ipsec proposal ipsec_prop]
user@host# set lifetime-seconds 3600
```

7. Create the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec_pol
```

8. Specify the IPsec proposal reference.

```
[edit security ipsec policy ipsec_pol]
user@host# set proposals ipsec_prop
```

9. Specify the interface to bind.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 bind-interface st0.0
```

10. Configure the forwarding class to the multiple IPsec SA.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class ef-class
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class af-class
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class res-class
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class web-data
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class control-data
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class voip-data
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class network-control
user@host# set vpn ipsec_vpn1 multi-sa forwarding-class best-effort
```

11. Specify the IKE gateway.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 ike gateway gw_sunnyvale
```

12. Specify the IPsec policies.

```
[edit security ipsec]
user@host# set vpn ipsec_vpn1 ike ipsec-policy ipsec_pol
```

13. Specify that the tunnel be brought up immediately to negotiate IPsec SA when the first data packet arrives to be sent.

    ```
    [edit security ipsec]
    user@host# set vpn ipsec_vpn1 establish-tunnels immediately
    ```

14. Configure local IP addresses for a traffic selector.

    ```
    [edit security ipsec]
    user@host# set vpn ipsec_vpn1 traffic-selector ipsec_vpn1_TS1 local-ip 203.0.113.2/25
    ```

15. Configure remote IP addresses for a traffic selector.

    ```
    [edit security ipsec]
    user@host# set vpn ipsec_vpn1 traffic-selector ipsec_vpn1_TS1 remote-ip 192.0.2.30/24
    ```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ipsec
traceoptions {
    flag all;
}
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha-256;
    encryption-algorithm aes256-cbc;
}
proposal ipsec_prop {
    lifetime-seconds 3600;
}
policy ipsec_pol {
    proposals ipsec_prop;
}
vpn ipsec_vpn1 {
```

```
    bind-interface st0.0;
    multi-sa {
        forwarding-class ef-class;
        forwarding-class af-class;
        forwarding-class res-class;
        forwarding-class web-data;
        forwarding-class control-data;
        forwarding-class voip-data;
        forwarding-class network-control;
        forwarding-class best-effort;
    }
    ike {
        gateway gw_sunnyvale;
        ipsec-policy ipsec_pol;
    }
    traffic-selector ipsec_vpn1_TS1 {
        local-ip 203.0.113.2/25;
        remote-ip 192.0.2.30/24;
    }
    establish-tunnels immediately;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Security Policies

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security policies from-zone trust to-zone vpn policy vpn match source-address sunnyvale
set security policies from-zone trust to-zone vpn policy vpn match destination-address chicago
set security policies from-zone trust to-zone vpn policy vpn match application any
set security policies from-zone trust to-zone vpn policy vpn then permit
set security policies from-zone vpn to-zone trust policy vpn match source-address chicago
set security policies from-zone vpn to-zone trust policy vpn match destination-address sunnyvale
set security policies from-zone vpn to-zone trust policy vpn match application any
set security policies from-zone vpn to-zone trust policy vpn then permit
```

Enable security policies trace options for troubleshooting the policy-related issues.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure security policies:

1. Create the security policy to permit traffic from the trust zone to the vpn zone.

```
[edit security policies from-zone trust to-zone vpn]
user@host# set policy vpn match source-address sunnyvale
user@host# set policy vpn match destination-address chicago
user@host# set policy vpn match application any
user@host# set policy vpn then permit
```

2. Create the security policy to permit traffic from the vpn zone to the trust zone.

```
[edit security policies from-zone vpn to-zone trust]
user@host# set policy vpn match source-address chicago
user@host# set policy vpn match destination-address sunnyvale
user@host# set policy vpn match application any
user@host# set policy vpn then permit
```

Results

From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone trust to-zone vpn {
    policy vpn {
        match {
            source-address sunnyvale;
            destination-address chicago;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone vpn to-zone trust {
    policy vpn {
        match {
            source-address chicago;
            destination-address sunnyvale;
            application any;
        }
        then {
            permit;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying IPsec Security Associations

Purpose

Verify the IPsec status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *`131073`* `detail` and `show security ipsec statistics index` *`131073`* commands.

For brevity, the show command outputs does not display all the values of the configuration. Only a subset of the configuration is displayed. Rest of the configuration on the system has been replaced with ellipses (...).

```
user@host> show security ipsec security-associations
Total active tunnels: 2     Total Ipsec sas: 18
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131073 ESP:aes256/sha256 2d8e710b 1949/ unlim   -   root 500   5.0.0.1
  >131073 ESP:aes256/sha256 5f3a3239 1949/ unlim   -   root 500   5.0.0.1
  <131073 ESP:aes256/sha256 5d227e19 1949/ unlim   -   root 500   5.0.0.1
  >131073 ESP:aes256/sha256 5490da   1949/ unlim   -   root 500   5.0.0.1
  <131073 ESP:aes256/sha256 211fb8bc 1949/ unlim   -   root 500   5.0.0.1
  >131073 ESP:aes256/sha256 dde29cd0 1949/ unlim   -   root 500   5.0.0.1
  <131073 ESP:aes256/sha256 49b64080 1949/ unlim   -   root 500   5.0.0.1
  >131073 ESP:aes256/sha256 314afea0 1949/ unlim   -   root 500   5.0.0.1
  <131073 ESP:aes256/sha256 fec6f6ea 1949/ unlim   -   root 500   5.0.0.1
  >131073 ESP:aes256/sha256 428a3a0d 1949/ unlim   -   root 500   5.0.0.1
...
```

```
user@host> show security ipsec security-associations index 131073 detail

ID: 131073 Virtual-system: root, VPN Name: IPSEC_VPN1
  Local Gateway: 4.0.0.1, Remote Gateway: 5.0.0.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0
  Port: 500, Nego#: 18, Fail#: 0, Def-Del#: 0 Flag: 0x600a39
  Multi-sa, Configured SAs# 9, Negotiated SAs#: 9
  Tunnel events:
    Mon Apr 23 2018 22:20:54 -0700: IPSec SA negotiation successfully completed (1 times)
    Mon Apr 23 2018 22:20:54 -0700: IKE SA negotiation successfully completed (2 times)
    Mon Apr 23 2018 22:20:18 -0700: User cleared IKE SA from CLI, corresponding IPSec SAs
cleared (1 times)
    Mon Apr 23 2018 22:19:55 -0700: IPSec SA negotiation successfully completed (2 times)
    Mon Apr 23 2018 22:19:23 -0700: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Mon Apr 23 2018 22:19:23 -0700: Bind-interface's zone received. Information updated (1 times)
    Mon Apr 23 2018 22:19:23 -0700: External interface's zone received. Information updated (1
times)
  Direction: inbound, SPI: 2d8e710b, AUX-SPI: 0
```

```
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1563 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Multi-sa FC Name: default
  Direction: outbound, SPI: 5f3a3239, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1563 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Multi-sa FC Name: default
  Direction: inbound, SPI: 5d227e19, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1551 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Multi-sa FC Name: best-effort
  Direction: outbound, SPI: 5490da, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1551 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
...
```

```
user@host> show security ipsec statistics index 131073

ESP Statistics:
  Encrypted bytes:              952
  Decrypted bytes:              588
  Encrypted packets:              7
  Decrypted packets:              7
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
FC Name     Encrypted Pkts  Decrypted Pkts  Encrypted bytes  Decrypted bytes
  best-effort 7                 7                 952                588
  custom_q1   0                 0                 0                  0
  custom_q2   0                 0                 0                  0
  network-control 0             0                 0                  0
  custom_q4   0                 0                 0                  0
  custom_q5   0                 0                 0                  0
  custom_q6   0                 0                 0                  0
  custom_q7   0                 0                 0                  0
  default     0                 0                 0                  0
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 131073. Use this value with the show security ipsec security-associations index command to get more information about this particular SA.

- There is one IPsec SA pair using port 500.

- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 1949/ unlim value indicates that the Phase lifetime expires in 1949 seconds, and that no lifesize has been specified, which indicates that it is unlimited.

- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.

The `show security ike security-associations index 131073 detail` command lists additional information about the SA with an index number of 131073:

- The local identity and remote identity make up the proxy ID for the SA. A proxy ID mismatch is one of the most common causes for a Phase failure. If no IPsec SA is listed, confirm that Phase proposals, including the proxy ID settings, are correct for both peers.

- Displays all the child SA details including forwarding class name.

The `show security ipsec statistics index 131073` command lists statistics for each forwarding class name.

- An error value of zero in the output indicates a normal condition.
- We recommend running this command multiple times to observe any packet loss issues across a VPN. Output from this command also displays the statistics for encrypted and decrypted packet counters, error counters, and so on.
- You must enable security flow trace options to investigate which ESP packets are experiencing errors and why.

SEE ALSO

Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs

Understanding Traffic Selectors and CoS-Based IPsec VPNs

IPsec Overview | 20

Example: Configuring a Policy-Based VPN | 268

Introduction to IKE in Junos OS | 136

Understanding CoS Support on st0 Interfaces

IN THIS SECTION

Starting with Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, class of service (CoS) features such as classifier, policer, queuing, scheduling, shaping, rewriting markers, and virtual channels can now be configured on the secure tunnel interface (st0) for point-to-point VPNs.

The st0 tunnel interface is an internal interface that can be used by route-based VPNs to route cleartext traffics to an IPsec VPN tunnel. The following CoS features are supported on the st0 interface on all available SRX Series Firewalls and vSRX2.0:

- Classifiers
- Policers

- Queuing, scheduling, and shaping
- Rewrite markers
- Virtual channels

Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, support for queuing, scheduling, shaping, and virtual channels is added to the st0 interface for SRX5400, SRX5600, and SRX5800 devices. Support for all the listed CoS features is added for the st0 interface for SRX1500, SRX4100, and SRX4200 devices. Starting with Junos OS Release 17.4R1, support for listed CoS features is added for the st0 interface for SRX4600 devices.

Limitations of CoS support on VPN st0 interfaces

The following limitations apply to CoS support on VPN st0 interfaces:

- The maximum number for software queues is 2048. If the number of st0 interfaces exceeds 2048, not enough software queues can be created for all the st0 interfaces.
- Only route-based VPNs can apply CoS features on st0 interfaces. Table 66 on page 589 describes the st0 CoS feature support for different types of VPNs.

Table 66: CoS Feature Support for VPN

Classifier Features	Site-to-Site VPN (P2P)	AutoVPN (P2P)	Site-to-Site/Auto VPN /AD-VPN (P2MP)
Classifiers, policers, and rewriting markers	Supported	Supported	Supported
Queueing, scheduling, and shaping based on st0 logical interfaces	Supported	Not supported	Not supported
Queueing, scheduling, and shaping based on virtual channels	Supported	Supported	Supported

- On SRX300, SRX320, SRX340, SRX345, and SRX550HM devices, one st0 logical interface can bind to multiple VPN tunnels. The eight queues for the st0 logical interface cannot reroute the traffic to different tunnels, so pre-tunneling is not supported.

  The virtual channel feature can be used as a workaround on SRX300, SRX320, SRX340, SRX345, and SRX550HM devices.

- When defining a CoS shaping rate on an st0 tunnel interface, consider the following restrictions:

  - The shaping rate on the tunnel interface must be less than that of the physical egress interface.

  - The shaping rate only measures the packet size that includes the inner Layer 3 cleartext packet with an ESP/AH header and an outer IP header encapsulation. The outer Layer 2 encapsulation added by the physical interface is not factored into the shaping rate measurement.

  - The CoS behavior works as expected when the physical interface carries the shaped GRE or IP-IP tunnel traffic only. If the physical interface carries other traffic, thereby lowering the available bandwidth for tunnel interface traffic, the CoS features do not work as expected.

- On SRX550M, SRX5400, SRX5600, and SRX5800 devices, bandwidth limit and burst size limit values in a policer configuration are a per-SPU, not per-system limitation. This is the same policer behavior as on the physical interface.

SEE ALSO

> Class of Service User Guide (Security Devices)

Release History Table

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, support for listed CoS features is added for the st0 interface for SRX4600 devices.
15.1X49-D70	Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, support for queuing, scheduling, shaping, and virtual channels is added to the st0 interface for SRX5400, SRX5600, and SRX5800 devices. Support for all the listed CoS features is added for the st0 interface for SRX1500, SRX4100, and SRX4200 devices.
15.1X49-D60	Starting with Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, class of service (CoS) features such as classifier, policer, queuing, scheduling, shaping, rewriting markers, and virtual channels can now be configured on the secure tunnel interface (st0) for point-to-point VPNs.

RELATED DOCUMENTATION

Class of Service User Guide (Security Devices)

9

CHAPTER

NAT-T

Route-Based and Policy-Based VPNs with NAT-T

IN THIS SECTION

Network Address Translation-Traversal (NAT-T) is a method used for managing IP address translation-related issues encountered when the data protected by IPsec passes through a device configured with NAT for address translation.

Understanding NAT-T

Network Address Translation-Traversal (NAT-T) is a method for getting around IP address translation issues encountered when data protected by IPsec passes through a NAT device for address translation. Any changes to the IP addressing, which is the function of NAT, causes IKE to discard packets. After detecting one or more NAT devices along the datapath during Phase 1 exchanges, NAT-T adds a layer of User Datagram Protocol (UDP) encapsulation to IPsec packets so they are not discarded after address translation. NAT-T encapsulates both IKE and ESP traffic within UDP with port 4500 used as both the source and destination port. Because NAT devices age out stale UDP translations, keepalive messages are required between the peers.

NAT-T is enabled by default therefore you must use the `no-nat-traversal` statement at the `[edit security ike gateway` *`gateway-name`* hierarchy level for disabling the NAT-T.

There are two broad categories of NAT:

- Static NAT, where there is a one-to-one relationship between the private and public addresses. Static NAT works in both inbound and outbound directions.
- Dynamic NAT, where there is a many-to-one or many-to-many relationship between the private and public addresses. Dynamic NAT works in the outbound direction only.

The location of a NAT device can be such that:

- Only the IKEv1 or IKEv2 initiator is behind a NAT device. Multiple initiators can be behind separate NAT devices. Initiators can also connect to the responder through multiple NAT devices.
- Only the IKEv1 or IKEv2 responder is behind a NAT device.
- Both the IKEv1 or IKEv2 initiator and the responder are behind a NAT device.

Dynamic endpoint VPN covers the situation where the initiator's IKE external address is not fixed and is therefore not known by the responder. This can occur when the initiator's address is dynamically assigned by an ISP or when the initiator's connection crosses a dynamic NAT device that allocates addresses from a dynamic address pool.

Configuration examples for NAT-T are provided for the topology in which only the responder is behind a NAT device and the topology in which both the initiator and responder are behind a NAT device. Site-to-site IKE gateway configuration for NAT-T is supported on both the initiator and responder. A remote IKE ID is used to validate a peer’s local IKE ID during Phase 1 of IKE tunnel negotiation. Both the initiator and responder require a `local-identity` and a `remote-identity` setting.

On SRX5400, SRX5600, and SRX5800 devices, the IPsec NAT-T tunnel scaling and sustaining issues are as follows:

- For a given private IP address, the NAT device should translate both 500 and 4500 private ports to the same public IP address.
- The total number of tunnels from a given public translated IP cannot exceed 1000 tunnels.

Starting from Junos OS Release 19.2R1, PowerMode IPSec (PMI) for NAT-T is supported only on SRX5400, SRX5600, and SRX5800 devices equipped with SRX5K-SPC3 Services Processing Card (SPC), or with vSRX Virtual Firewall.

SEE ALSO

IPsec Overview | 20

Example: Configuring a Route-Based VPN with the Responder behind a NAT Device

IN THIS SECTION

This example shows how to configure a route-based VPN with a responder behind a NAT device between a branch office and the corporate office.

Requirements

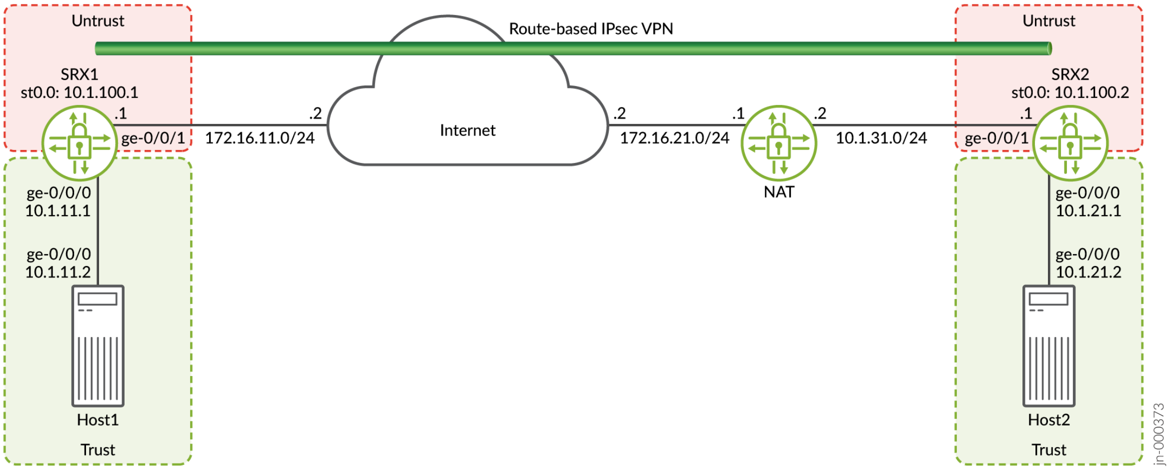
Before you begin, read "IPsec Overview" on page 20.

Overview

In this example, you configure a route-based VPN. Host1 will use the VPN to connect to their corporate headquarters on SRX2.

Figure 46 on page 595 shows an example of a topology for route-based VPN with only the responder behind a NAT device.

Figure 46: Route-Based VPN Topology with Only the Responder behind a NAT Device



In this example, you configure interfaces, IPsec, and security policies for both an initiator in SRX1 and a responder in SRX2. Then you configure IKE Phase 1 and IPsec Phase 2 parameters.

SRX1 sends packets with the destination address of 172.16.21.1 to establish the VPN. The NAT device translates the destination address to 10.1.31.1.

See Table 67 on page 596 through Table 69 on page 598 for specific configuration parameters used for the initiator in the examples.

Table 67: Interface, Routing Options, and Security Parameters for SRX1

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/1	172.16.11.1/24
	ge-0/0/0	10.1.11.1/24
	st0.0 (tunnel interface)	10.1.100.1/24
Static routes	10.1.21.0/24	The next hop is st0.0.
	172.16.21.1/32	The next hop is 172.16.11.2.
Security zones	untrust	• The system services of IKE and ping. • The ge-0/0/1.0 and the st0.0 interfaces are bound to this zone.
	trust	• Allow all system services. • Allow all protocols. • The ge-0/0/0.0 interface is bound to this zone.
Security policies	to-SRX2	Permit traffic from 10.1.11.0/24 in the trust zone to 10.1.21.0/24 in the untrust zone.

Table 67: Interface, Routing Options, and Security Parameters for SRX1 *(Continued)*

Feature	Name	Configuration Parameters
	from-SRX2	Permit traffic from 10.1.21.0/24 in the untrust zone to 10.1.11.0/24 in the trust zone.

Table 68: IKE Phase 1 Configuration Parameters for SRX1

Feature	Name	Configuration Parameters
Proposal	ike_prop	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: 3des-cbc
Policy	ike_pol	• Mode: main • Proposal reference: ike_prop • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw1	• IKE policy reference: ike_pol • External interface: ge-0/0/1.0 • Gateway address: 172.16.21.1 • Local peer (initiator): branch_natt1@example.net • Remote peer (responder): responder_natt1@example.net

Table 69: IPsec Phase 2 Configuration Parameters for SRX1

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: 3des-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop • Perfect forward secrecy (PFS) keys: group2
VPN	vpn1	• IKE gateway reference: gw1 • IPsec policy reference: ipsec_pol • Bind to interface: st0.0 • Establish tunnels immediately

See Table 70 on page 598 through Table 72 on page 600 for specific configuration parameters used for the responder in the examples.

Table 70: Interface, Routing Options, and Security Parameters for SRX2

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/1	10.1.31.1/24
	ge-0/0/0	10.1.21.1/24
	st0.0 (tunnel interface)	10.1.100.2/24
Static routes	172.16.11.1/32	The next hop is 10.1.31.2.
	10.1.11.0/24	The next hop is st0.0.

Table 70: Interface, Routing Options, and Security Parameters for SRX2 *(Continued)*

Feature	Name	Configuration Parameters
Security zones	untrust	• Allow IKE and ping system services. • The ge-0/0/1.0 and the st0.0 interfaces are bound to this zone.
	trust	• Allow all system services. Allow all protocols. • The ge-0/0/0.0 interface is bound to this zone.
Security policies	to-SRX1	Permit traffic from 10.1.21.0/24 in the trust zone to 10.1.11.0/24 in the untrust zone.
	from-SRX1	Permit traffic from 10.1.11.0/24 in the untrust zone to 10.1.21.0/24 in the trust zone.

Table 71: IKE Phase 1 Configuration Parameters for SRX2

Feature	Name	Configuration Parameters
Proposal	ike_prop	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: sha1 • Encryption algorithm: 3des-cbc

Table 71: IKE Phase 1 Configuration Parameters for SRX2 *(Continued)*

Feature	Name	Configuration Parameters
Policy	ike_pol	• Mode: main • Proposal reference: ike_prop • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gw1	• IKE policy reference: ike_pol • External interface: ge-0/0/1.0 • Gateway address: 172.16.11.1 • Local peer (responder): responder_natt1@example.net • Remote peer (initiator): branch_natt1@example.net

Table 72: IPsec Phase 2 Configuration Parameters for SRX2

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-sha1-96 • Encryption algorithm: 3des-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop • PFS keys: group2
VPN	vpn1	• IKE gateway reference: gw1 • IPsec policy reference: ipsec_pol • Bind to interface: st0.0 • Establish tunnels immediately

Configuration

IN THIS SECTION

Configuring Interface, Routing Options, and Security Parameters for SRX1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security address-book book1 address Host1 10.1.11.0/24
set security address-book book1 attach zone trust
set security address-book book2 address Host2 10.1.21.0/24
set security address-book book2 attach zone untrust
set security policies from-zone trust to-zone untrust policy to-SRX2 match source-address Host1
set security policies from-zone trust to-zone untrust policy to-SRX2 match destination-address
Host2
set security policies from-zone trust to-zone untrust policy to-SRX2 match application any
set security policies from-zone trust to-zone untrust policy to-SRX2 then permit
set security policies from-zone untrust to-zone trust policy from-SRX2 match source-address Host2
set security policies from-zone untrust to-zone trust policy from-SRX2 match destination-address
Host1
set security policies from-zone untrust to-zone trust policy from-SRX2 match application any
set security policies from-zone untrust to-zone trust policy from-SRX2 then permit
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet address 10.1.11.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.11.1/24
set interfaces st0 unit 0 family inet address 10.1.100.1/24
set routing-options static route 10.1.21.0/24 next-hop st0.0
set routing-options static route 172.16.21.1/32 next-hop 172.16.11.2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure interfaces, static routes, and security parameters:

1. Configure the interfaces connected to the Internet, Host1, and the interface used for the VPN.

```
[edit]
user@SRX1# set interfaces ge-0/0/0 unit 0 family inet address 10.1.11.1/24
user@SRX1# set interfaces ge-0/0/1 unit 0 family inet address 172.16.11.1/24
user@SRX1# set interfaces st0 unit 0 family inet address 10.1.100.1/24
```

2. Configure static routes for the traffic that will use the VPN and for SRX1 to reach the NAT device.

```
[edit]
user@SRX1# set routing-options static route 10.1.21.0/24 next-hop st0.0
user@SRX1# set routing-options static route 172.16.21.1/32 next-hop 172.16.11.2
```

3. Configure the untrust security zone.

```
[edit]
user@SRX1# set security zones security-zone untrust host-inbound-traffic system-services ike
user@SRX1# set security zones security-zone untrust host-inbound-traffic system-services ping
user@SRX1# set security zones security-zone untrust interfaces st0.0
user@SRX1# set security zones security-zone untrust interfaces ge-0/0/1.0
```

4. Configure the trust security zone.

```
[edit]
user@SRX1# set security zones security-zone trust host-inbound-traffic system-services all
user@SRX1# set security zones security-zone trust host-inbound-traffic protocols all
user@SRX1# set security zones security-zone trust interfaces ge-0/0/0.0
```

5. Configure address books for the networks used in the security policies.

```
[edit]
user@SRX1# set security address-book book1 address Host1 10.1.11.0/24
user@SRX1# set security address-book book1 attach zone trust
user@SRX1# set security address-book book2 address Host2 10.1.21.0/24
user@SRX1# set security address-book book2 attach zone untrust
```

6. Create security policies to allow traffic between the hosts.

```
[edit]
user@SRX1# set security policies from-zone trust to-zone untrust policy to-SRX2 match source-address Host1
user@SRX1# set security policies from-zone trust to-zone untrust policy to-SRX2 match destination-address Host2
user@SRX1# set security policies from-zone trust to-zone untrust policy to-SRX2 match application any
user@SRX1# set security policies from-zone trust to-zone untrust policy to-SRX2 then permit
user@SRX1# set security policies from-zone untrust to-zone trust policy from-SRX2 match source-address Host2
user@SRX1# set security policies from-zone untrust to-zone trust policy from-SRX2 match destination-address Host1
user@SRX1# set security policies from-zone untrust to-zone trust policy from-SRX2 match application any
user@SRX1# set security policies from-zone untrust to-zone trust policy from-SRX2 then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.1.11.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 172.16.11.1/24;
        }
    }
}
st0 {
    unit 0 {
        family inet {
            address 10.1.100.1/24;
        }
    }
}
```

```
[edit]
user@SRX1# show routing-options
static {
    route 10.1.21.0/24 next-hop st0.0;
    route 172.16.21.1/32 next-hop 172.16.11.2;
}
```

```
[edit]
user@SRX1# show security
```

```
address-book {
    book1 {
        address Host1 10.1.11.0/24;
        attach {
            zone trust;
        }
    }
    book2 {
        address Host2 10.1.21.0/24;
        attach {
            zone untrust;
        }
    }
}
policies {
    from-zone trust to-zone untrust {
        policy to-SRX2 {
            match {
                source-address Host1;
                destination-address Host2;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone untrust to-zone trust {
        policy from-SRX2 {
            match {
                source-address Host2;
                destination-address Host1;
                application any;
            }
            then {
                permit;
            }
        }
    }
}
zones {
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                ike;
                ping;
            }
        }
        interfaces {
            st0.0;
            ge-0/0/1.0;
        }
    }
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE for SRX1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike_prop authentication-method pre-shared-keys
set security ike proposal ike_prop dh-group group2
set security ike proposal ike_prop authentication-algorithm sha1
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_pol mode main
set security ike policy ike_pol proposals ike_prop
set security ike policy ike_pol pre-shared-key ascii-text “$ABC123”
set security ike gateway gw1 ike-policy ike_pol
set security ike gateway gw1 address 172.16.21.1
set security ike gateway gw1 local-identity user-at-hostname "srx1@example.com"
set security ike gateway gw1 remote-identity user-at-hostname "srx2@example.com"
set security ike gateway gw1 external-interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE:

1. Create an IKE Phase 1 proposal.

```
[edit]
user@SRX1# set security ike proposal ike_prop authentication-method pre-shared-keys
user@SRX1# set security ike proposal ike_prop dh-group group2
user@SRX1# set security ike proposal ike_prop authentication-algorithm sha1
user@SRX1# set security ike proposal ike_prop encryption-algorithm 3des-cbc
```

2. Create an IKE Phase 1 policy.

```
[edit]
user@SRX1# set security ike policy ike_pol mode main
user@SRX1# set security ike policy ike_pol proposals ike_prop
user@SRX1# set security ike policy ike_pol pre-shared-key ascii-text “$ABC123”
```

3. Configure the IKE Phase 1 gateway parameters. The gateway address should be the IP for the NAT device.

```
[edit security ike gateway gw1]
user@SRX1# set security ike gateway gw1 ike-policy ike_pol
user@SRX1# set security ike gateway gw1 address 172.16.21.1
user@SRX1# set security ike gateway gw1 local-identity user-at-hostname "srx1@example.com"
user@SRX1# set security ike gateway gw1 remote-identity user-at-hostname "srx2@example.com"
user@SRX1# set security ike gateway gw1 external-interface ge-0/0/1.0
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX1# show security ike
proposal ike_prop {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
}
policy ike_pol {
    mode main;
    proposals ike_prop;
    pre-shared-key ascii-text “$9$xPn7-VwsgaJUHqp01IcSs2g”; ## SECRET-DATA
}
gateway gw1 {
    ike-policy ike_pol;
    address 172.16.21.1;
    local-identity user-at-hostname "srx1@example.com";
    remote-identity user-at-hostname "srx2@example.com";
    external-interface ge-0/0/1.0;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec for SRX1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec policy ipsec_pol perfect-forward-secrecy keys group2
set security ipsec policy ipsec_pol proposals ipsec_prop
set security ipsec vpn vpn1 bind-interface st0.0
set security ipsec vpn vpn1 ike gateway gw1
set security ipsec vpn vpn1 ike ipsec-policy ipsec_pol
set security ipsec vpn vpn1 establish-tunnels immediately
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@SRX1# set security ipsec proposal ipsec_prop protocol esp
user@SRX1# set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
user@SRX1# set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
```

2. Create the IPsec Phase 2 policy.

```
[edit]
user@SRX1# set security ipsec policy ipsec_pol perfect-forward-secrecy keys group2
user@SRX1# set security ipsec policy ipsec_pol proposals ipsec_prop
```

3. Configure the IPsec VPN parameters.

```
[edit]
user@SRX1# set security ipsec vpn vpn1 bind-interface st0.0
user@SRX1# set security ipsec vpn vpn1 ike gateway gw1
user@SRX1# set security ipsec vpn vpn1 ike ipsec-policy ipsec_pol
user@SRX1# set security ipsec vpn vpn1 establish-tunnels immediately
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX1# show security ipsec
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
}
policy ipsec_pol {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec_prop;
}
vpn vpn1 {
    bind-interface st0.0;
    ike {
        gateway gw1;
        ipsec-policy ipsec_pol;
    }
    establish-tunnels immediately;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Interfaces, Routing Options, and Security Parameters for SRX2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security address-book book1 address Host2 10.1.21.0/24
set security address-book book1 attach zone trust
set security address-book book2 address Host1 10.1.11.0/24
set security address-book book2 attach zone untrust
set security policies from-zone trust to-zone untrust policy to-SRX1 match source-address Host2
set security policies from-zone trust to-zone untrust policy to-SRX1 match destination-address
Host1
set security policies from-zone trust to-zone untrust policy to-SRX1 match application any
set security policies from-zone trust to-zone untrust policy to-SRX1 then permit
set security policies from-zone untrust to-zone trust policy from-SRX1 match source-address Host1
set security policies from-zone untrust to-zone trust policy from-SRX1 match destination-address
Host2
set security policies from-zone untrust to-zone trust policy from-SRX1 match application any
set security policies from-zone untrust to-zone trust policy from-SRX1 then permit
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet address 10.1.21.1/24
set interfaces ge-0/0/1 unit 0 family inet address 10.1.31.1/24
set interfaces st0 unit 0 family inet address 10.1.100.2/24
set routing-options static route 172.16.11.1/32 next-hop 10.1.31.2
set routing-options static route 10.1.11.0/24 next-hop st0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure interfaces, static routes, and security parameters:

1. Configure the interfaces connected to the Internet, Host2, and the interface used for the VPN.

   ```
   [edit]
   user@SRX2# set interfaces ge-0/0/0 unit 0 family inet address 10.1.21.1/24
   user@SRX2# set interfaces ge-0/0/1 unit 0 family inet address 10.1.31.1/24
   user@SRX2# set interfaces st0 unit 0 family inet address 10.1.100.2/24
   ```

2. Configure static routes for the traffic that will use the VPN and for SRX2 to reach SRX1.

```
[edit]
user@SRX2# set routing-options static route 172.16.11.1/32 next-hop 10.1.31.2
user@SRX2# set routing-options static route 10.1.11.0/24 next-hop st0.0
```

3. Configure the untrust security zone.

```
[edit]
user@SRX2# set security zones security-zone untrust host-inbound-traffic system-services ike
user@SRX2# set security zones security-zone untrust host-inbound-traffic system-services ping
user@SRX2# set security zones security-zone untrust interfaces ge-0/0/1.0
user@SRX2# set security zones security-zone untrust interfaces st0.0
```

4. Configure the trust security zone.

```
[edit]
user@SRX2# set security zones security-zone trust host-inbound-traffic system-services all
user@SRX2# set security zones security-zone trust host-inbound-traffic protocols all
user@SRX2# set security zones security-zone trust interfaces ge-0/0/0.0
```

5. Configure address books for the networks used in the security policies.

```
[edit]
user@SRX2# set security address-book book1 address Host2 10.1.21.0/24
user@SRX2# set security address-book book1 attach zone trust
user@SRX2# set security address-book book2 address Host1 10.1.11.0/24
user@SRX2# set security address-book book2 attach zone untrust
```

6. Create security policies to allow traffic between the hosts.

```
[edit]
user@SRX2# set security policies from-zone trust to-zone untrust policy to-SRX1 match source-
address Host2
user@SRX2# set security policies from-zone trust to-zone untrust policy to-SRX1 match
destination-address Host1
user@SRX2# set security policies from-zone trust to-zone untrust policy to-SRX1 match
application any
```

```
user@SRX2# set security policies from-zone trust to-zone untrust policy to-SRX1 then permit
user@SRX2# set security policies from-zone untrust to-zone trust policy from-SRX1 match
source-address Host1
user@SRX2# set security policies from-zone untrust to-zone trust policy from-SRX1 match
destination-address Host2
user@SRX2# set security policies from-zone untrust to-zone trust policy from-SRX1 match
application any
user@SRX2# set security policies from-zone untrust to-zone trust policy from-SRX1 then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX2# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.1.21.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.31.1/24;
        }
    }
}
st0 {
    unit 0 {
        family inet {
            address 10.1.100.2/24;
        }
    }
}
```

```
[edit]
user@SRX2# show routing-options
static {
    route 172.16.11.1/32 next-hop 10.1.31.2;
    route 10.1.11.0/24 next-hop st0.0;
}
```

```
[edit]
user@SRX2# show security
address-book {
    book1 {
        address Host2 10.1.21.0/24;
        attach {
            zone trust;
        }
    }
    book2 {
        address Host1 10.1.11.0/24;
        attach {
            zone untrust;
        }
    }
}
policies {
    from-zone trust to-zone untrust {
        policy to-SRX1 {
            match {
                source-address Host2;
                destination-address Host1;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone untrust to-zone trust {
        policy from-SRX1 {
            match {
                source-address Host1;
                destination-address Host2;
                application any;
            }
            then {
                permit;
            }
        }
    }
}
zones {
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                ike;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
            st0.0;
        }
    }
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE for SRX2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike_prop authentication-method pre-shared-keys
set security ike proposal ike_prop dh-group group2
set security ike proposal ike_prop authentication-algorithm sha1
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_pol mode main
set security ike policy ike_pol proposals ike_prop
set security ike policy ike_pol pre-shared-key ascii-text “$ABC123”
set security ike gateway gw1 ike-policy ike_pol
set security ike gateway gw1 address 172.16.11.1
set security ike gateway gw1 local-identity user-at-hostname "srx2@example.com"
set security ike gateway gw1 remote-identity user-at-hostname "srx1@example.com"
set security ike gateway gw1 external-interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE:

1. Create an IKE Phase 1 proposal.

```
[edit]
user@SRX2# set security ike proposal ike_prop authentication-method pre-shared-keys
user@SRX2# set security ike proposal ike_prop dh-group group2
user@SRX2# set security ike proposal ike_prop authentication-algorithm sha1
user@SRX2# set security ike proposal ike_prop encryption-algorithm 3des-cbc
```

2. Create an IKE Phase 1 policy.

```
[edit]
user@SRX2# set security ike policy ike_pol mode main
```

```
user@SRX2# set security ike policy ike_pol proposals ike_prop
user@SRX2# set security ike policy ike_pol pre-shared-key ascii-text “$ABC123”
```

3. Configure the IKE Phase 1 gateway parameters. The gateway address should be the IP for SRX1.

```
[edit]
user@SRX2# set security ike gateway gw1 ike-policy ike_pol
user@SRX2# set security ike gateway gw1 address 172.16.11.1
user@SRX2# set security ike gateway gw1 local-identity user-at-hostname "srx2@example.com"
user@SRX2# set security ike gateway gw1 remote-identity user-at-hostname "srx1@example.com"
user@SRX2# set security ike gateway gw1 external-interface ge-0/0/1.0
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX2# show security ike
proposal ike_prop {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
}
policy ike_pol {
    mode main;
    proposals ike_prop;
    pre-shared-key ascii-text "$9$mP5QF3/At0IE-VsYoa36/"; ## SECRET-DATA
}
gateway gw1 {
    ike-policy ike_pol;
    address 172.16.11.1;
    local-identity user-at-hostname "srx2@example.com";
    remote-identity user-at-hostname "srx1@example.com";
    external-interface ge-0/0/1.0;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec for SRX2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec policy ipsec_pol perfect-forward-secrecy keys group2
set security ipsec policy ipsec_pol proposals ipsec_prop
set security ipsec vpn vpn1 bind-interface st0.0
set security ipsec vpn vpn1 ike gateway gw1
set security ipsec vpn vpn1 ike ipsec-policy ipsec_pol
set security ipsec vpn vpn1 establish-tunnels immediately
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

   ```
   [edit]
   user@SRX2# set security ipsec proposal ipsec_prop protocol esp
   user@SRX2# set security ipsec proposal ipsec_prop authentication-algorithm hmac-sha1-96
   user@SRX2# set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
   ```

2. Create the IPsec Phase 2 policy.

   ```
   [edit]
   user@SRX2# set security ipsec policy ipsec_pol perfect-forward-secrecy keys group2
   user@SRX2# set security ipsec policy ipsec_pol proposals ipsec_prop
   ```

3. Configure the IPsec VPN parameters.

```
[edit]
user@SRX2# set security ipsec vpn vpn1 bind-interface st0.0
user@SRX2# set security ipsec vpn vpn1 ike gateway gw1
user@SRX2# set security ipsec vpn vpn1 ike ipsec-policy ipsec_pol
user@SRX2# set security ipsec vpn vpn1 establish-tunnels immediately
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@SRX2# show security ipsec
proposal ipsec_prop {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
}
policy ipsec_pol {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec_prop;
}
vpn vpn1 {
    bind-interface st0.0;
    ike {
        gateway gw1;
        ipsec-policy ipsec_pol;
    }
    establish-tunnels immediately;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuration for the NAT Device

CLI Quick Configuration

Static NAT is used in the example. Static NAT is bidirectional which means that traffic from 10.1.31.1 to 172.16.11.1 will also use the same NAT configuration.

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security nat static rule-set rule1 from zone untrust
set security nat static rule-set rule1 rule ipsec match source-address 172.16.11.1/32
set security nat static rule-set rule1 rule ipsec match destination-address 172.16.21.1/32
set security nat static rule-set rule1 rule ipsec then static-nat prefix 10.1.31.1/32
set security policies from-zone trust to-zone untrust policy allow-out match source-address any
set security policies from-zone trust to-zone untrust policy allow-out match destination-address 
any
set security policies from-zone trust to-zone untrust policy allow-out match application any
set security policies from-zone trust to-zone untrust policy allow-out then permit
set security policies from-zone untrust to-zone trust policy allow-out-in match source-address 
any
set security policies from-zone untrust to-zone trust policy allow-out-in match destination-
address any
set security policies from-zone untrust to-zone trust policy allow-out-in match application any
set security policies from-zone untrust to-zone trust policy allow-out-in then permit
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet address 172.16.21.1/24
set interfaces ge-0/0/1 unit 0 family inet address 10.1.31.2/24
set routing-options static route 172.16.11.0/24 next-hop 172.16.21.2
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the IKE Phase 1 Status on SRX1

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the **show security ike security-associations** command. For a more detailed output, use the **show security ike security-associations detail** command.

```
user@SRX1> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
302301  UP     84e8fc61d0750278  ea9a07ef032805b6  Main           172.16.21.1
```

```
user@SRX1> show security ike security-associations detail
IKE peer 172.16.21.1, Index 302301, Gateway Name: gw1
  Role: Initiator, State: UP
  Initiator cookie: 84e8fc61d0750278, Responder cookie: ea9a07ef032805b6
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 172.16.11.1:4500, Remote: 172.16.21.1:4500
  Lifetime: Expires in 19657 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
  Remote Access Client Info: Unknown Client
  Peer ike-id: srx2@example.com
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
```

```
    Pseudo random function: hmac-sha1
    Diffie-Hellman group  : DH-group-2
  Traffic statistics:
    Input  bytes  :                 1780
    Output bytes  :                 2352
    Input  packets:                    7
    Output packets:                   14
    Input  fragmentated packets:       0
    Output fragmentated packets:       0
  IPSec security associations: 4 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 172.16.11.1:4500, Remote: 172.16.21.1:4500
    Local identity: srx1@example.com
    Remote identity: srx2@example.com
    Flags: IKE SA is created
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.
- Remote address—Verify that the remote IP address is correct and that port 4500 is being used for peer-to-peer communication. Remember that NAT-T encapsulates both IKE and ESP traffic within UDP with port 4500.
- Role initiator state
  - Up—The Phase 1 SA is established.
  - Down—There was a problem establishing the Phase 1 SA.
  - Both peers in the IPsec SA pair are using port 4500.
  - Peer IKE ID—Verify the remote address is correct.
  - Local identity and remote identity—Verify these are correct.

- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations` command lists additional information about security associations:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information

  Troubleshooting is best performed on the peer using the responder role.
- Initiator and responder information
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

Verifying IPsec Security Associations on SRX1

Purpose

Verify the IPsec status.

Action

From operational mode, enter the **show security ipsec security-associations** command. For a more detailed output, use the **show security ipsec security-associations detail** command.

```
user@SRX1> show security ipsec security-associations
  Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131073 ESP:3des/sha1   fc5dbac4 2160/ unlim  -   root 4500  172.16.21.1
  >131073 ESP:3des/sha1   45fed9d8 2160/ unlim  -   root 4500  172.16.21.1
```

```
user@SRX1> show security ipsec security-associations detail
ID: 131073 Virtual-system: root, VPN Name: vpn1
  Local Gateway: 172.16.11.1, Remote Gateway: 172.16.21.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0
  Port: 4500, Nego#: 7, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Fri Jul 22 2022 11:07:40 -0700: IPSec SA rekey successfully completed (3 times)
    Fri Jul 22 2022 08:38:41 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:38:41 -0700: User cleared IPSec SA from CLI (1 times)
    Fri Jul 22 2022 08:38:41 -0700: IKE SA negotiation successfully completed (3 times)
    Fri Jul 22 2022 08:38:26 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:38:26 -0700: User cleared IPSec SA from CLI (1 times)
    Fri Jul 22 2022 08:38:25 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:38:24 -0700: User cleared IPSec SA from CLI (1 times)
    Fri Jul 22 2022 08:37:37 -0700: IPSec SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: fc5dbac4, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2153 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1532 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 45fed9d8, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 2153 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1532 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The remote gateway has an address of 172.16.21.1.
- Both peers in the IPsec SA pair are using port 4500.
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 2160/ unlim value indicates that the Phase 2 lifetime expires in 2160 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.
- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

Verifying the IKE Phase 1 Status on SRX2

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the **show security ike security-associations** command. For a more detailed output, use the **show security ike security-associations detail** command.

```
user@SRX2> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
5567091 UP     84e8fc61d0750278  ea9a07ef032805b6  Main           172.16.11.1
```

```
user@SRX2> show security ike security-associations detail
IKE peer 172.16.11.1, Index 5567091, Gateway Name: gw1
  Role: Responder, State: UP
  Initiator cookie: 84e8fc61d0750278, Responder cookie: ea9a07ef032805b6
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.31.1:4500, Remote: 172.16.11.1:4500
  Lifetime: Expires in 18028 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
```

```
Remote Access Client Info: Unknown Client
Peer ike-id: srx1@example.com
AAA assigned IP: 0.0.0.0
Algorithms:
 Authentication        : hmac-sha1-96
 Encryption            : 3des-cbc
 Pseudo random function: hmac-sha1
 Diffie-Hellman group  : DH-group-2
Traffic statistics:
 Input  bytes  :                 2352
 Output bytes  :                 1780
 Input  packets:                   14
 Output packets:                    7
 Input  fragmentated packets:       0
 Output fragmentated packets:       0
IPSec security associations: 4 created, 3 deleted
Phase 2 negotiations in progress: 1

  Negotiation type: Quick mode, Role: Responder, Message ID: 0
  Local: 10.1.31.1:4500, Remote: 172.16.11.1:4500
  Local identity: srx2@example.com
  Remote identity: srx1@example.com
  Flags: IKE SA is created
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations detail` command to get more information about the SA.
- Remote address—Verify that the remote IP address is correct and that port 4500 is being used for peer-to-peer communication.
- Role responder state
  - Up—The Phase 1 SA has been established.
  - Down—There was a problem establishing the Phase 1 SA.

  - Peer IKE ID—Verify the address is correct.
  - Local identity and remote identity—Verify these addresses are correct.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations` command lists additional information about security associations:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information

  Troubleshooting is best performed on the peer using the responder role.
- Initiator and responder information
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

Verifying IPsec Security Associations on SRX2

Purpose

Verify the IPsec status.

Action

From operational mode, enter the **show security ipsec security-associations** command. For a more detailed output, use the **show security ipsec security-associations detail** command.

```
user@SRX2> show security ipsec security-associations
  Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131073 ESP:3des/sha1   45fed9d8 1526/ unlim  -   root 4500  172.16.11.1
  >131073 ESP:3des/sha1   fc5dbac4 1526/ unlim  -   root 4500  172.16.11.1
```

```
user@SRX2> show security ipsec security-associations detail
ID: 131073 Virtual-system: root, VPN Name: vpn1
  Local Gateway: 10.1.31.1, Remote Gateway: 172.16.11.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0
  Port: 4500, Nego#: 25, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Fri Jul 22 2022 11:07:40 -0700: IPSec SA negotiation successfully completed (4 times)
    Fri Jul 22 2022 08:38:41 -0700: Initial-Contact received from peer. Stale IKE/IPSec SAs
cleared (1 times)
    Fri Jul 22 2022 08:38:41 -0700: IKE SA negotiation successfully completed (5 times)
    Fri Jul 22 2022 08:38:26 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:38:26 -0700: IPSec SA delete payload received from peer, corresponding
IPSec SAs cleared (1 times)
    Fri Jul 22 2022 08:38:25 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:38:25 -0700: Initial-Contact received from peer. Stale IKE/IPSec SAs
cleared (1 times)
    Fri Jul 22 2022 08:37:37 -0700: IPSec SA negotiation successfully completed (1 times)
    Fri Jul 22 2022 08:37:37 -0700: IPSec SA delete payload received from peer, corresponding
IPSec SAs cleared (1 times)
    Thu Jul 21 2022 17:57:09 -0700: Peer's IKE-ID validation failed during negotiation (1 times)
    Thu Jul 21 2022 17:49:30 -0700: IKE SA negotiation successfully completed (4 times)
  Direction: inbound, SPI: 45fed9d8, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1461 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 885 seconds
```

```
      Mode: Tunnel(0 0), Type: dynamic, State: installed
      Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
      Anti-replay service: counter-based enabled, Replay window size: 64
    Direction: outbound, SPI: fc5dbac4, AUX-SPI: 0
                                 , VPN Monitoring: -
      Hard lifetime: Expires in 1461 seconds
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expires in 885 seconds
      Mode: Tunnel(0 0), Type: dynamic, State: installed
      Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
      Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The remote gateway has an ip address of 172.16.11.1.

- Both peers in the IPsec SA pair are using port 4500.

- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 1562/ unlim value indicates that the Phase 2 lifetime expires in 1562 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.

- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.

- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index` *`index_id`*`detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.

- Another common reason for Phase 2 failure is not specifying the ST interface binding. If IPsec cannot complete, check the kmd log or set trace options.

Verifying Host-to-Host Reachability

Purpose

Verify Host1 can reach Host2.

Action

From Host1 ping Host2. To verify the traffic is using the VPN, use the command `show security ipsec statistics` on SRX1. Clear the statistics by using the command `clear security ipsec statistics` before running the ping command.

```
user@Host1> ping 10.1.21.2 count 10 rapid
PING 10.1.21.2 (10.1.21.2): 56 data bytes
!!!!!!!!!!
--- 10.1.21.2 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.437/4.270/7.637/1.158 ms
```

```
user@SRX1> show security ipsec statistics
ESP Statistics:
  Encrypted bytes:            1360
  Decrypted bytes:             840
  Encrypted packets:            10
  Decrypted packets:            10
AH Statistics:
  Input bytes:                   0
  Output bytes:                  0
  Input packets:                 0
  Output packets:                0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

Meaning

The outputs show Host1 can ping Host2 and that the traffic is using the VPN.

SEE ALSO

IPsec Overview | 20

Example: Configuring a Policy-Based VPN

Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device

IN THIS SECTION

This example shows how to configure a policy-based VPN with both an initiator and a responder behind a NAT device to allow data to be securely transferred between a branch office and the corporate office.

Requirements

Before you begin, read "IPsec Overview" on page 20.

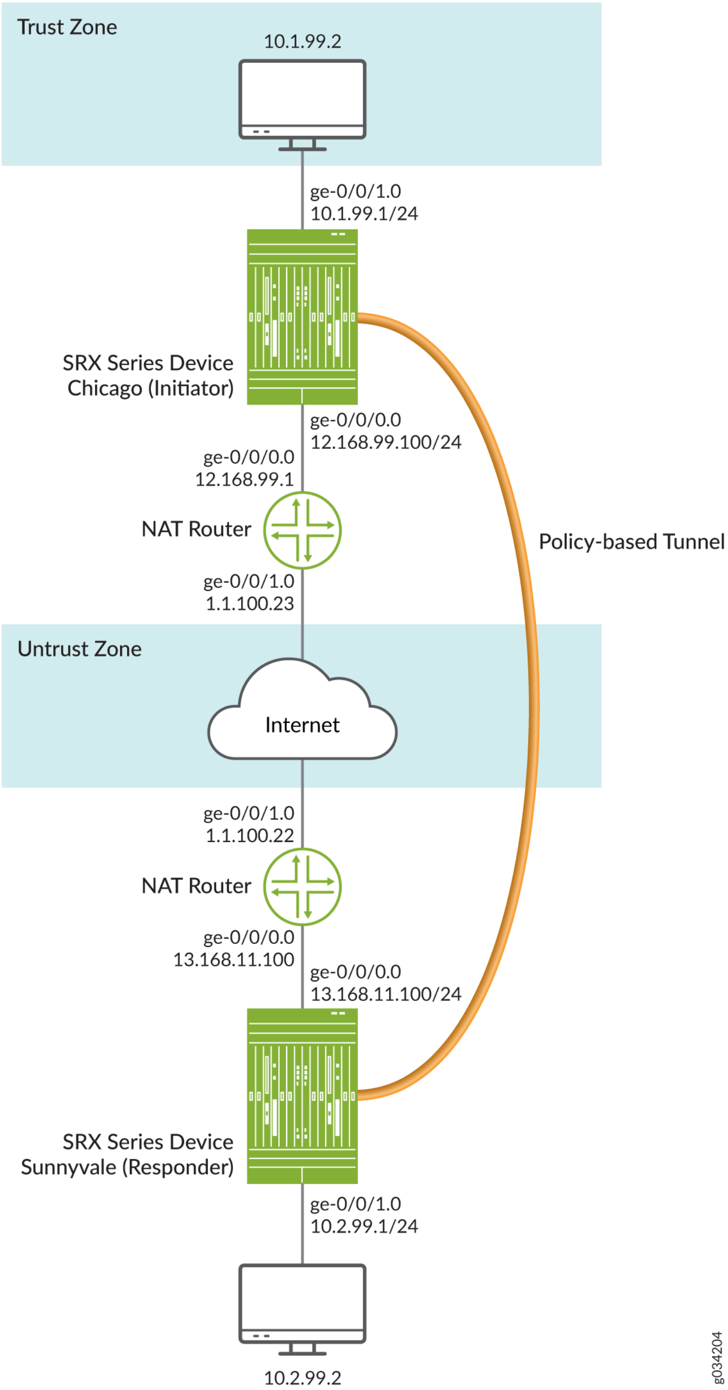
Overview

In this example, you configure a policy-based VPN for a branch office in Chicago, Illinois, because you want to conserve tunnel resources but still get granular restrictions on VPN traffic. Users in the branch office will use the VPN to connect to their corporate headquarters in Sunnyvale, California.

In this example, you configure interfaces, routing options, security zones, security policies for both an initiator and a responder.

Figure 47 on page 632 shows an example of a topology for a VPN with both an initiator and a responder behind a static NAT device.

Figure 47: Policy-Based VPN Topology with Both an Initiator and a Responder Behind a NAT Device



In this example, you configure interfaces, an IPv4 default route, and security zones. Then you configure IKE Phase 1, including local and remote peers, IPsec Phase 2, and the security policy. Note in the example above, the responder’s private IP address 13.168.11.1 is hidden by the static NAT device and mapped to public IP address 1.1.100.1.

See Table 73 on page 633 through Table 76 on page 635 for specific configuration parameters used for the initiator in the examples.

Table 73: Interface, Routing Options, and Security Zones for the Initiator

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0	12.168.99.100/24
	ge-0/0/1	10.1.99.1/24
Static routes	10.2.99.0/24 (default route)	The next hop is 12.168.99.100.
	1.1.100.0/24	12.168.99.100
Security zones	trust	• All system services are allowed. • All protocols are allowed. • The ge-0/0/1.0 interface is bound to this zone.
	untrust	• The ge-0/0/0.0 interface is bound to this zone.

Table 74: IKE Phase 1 Configuration Parameters for the Initiator

Feature	Name	Configuration Parameters
Proposal	ike_prop	• Authentication method: pre-shared-keys • Diffie-Hellman group: group2 • Authentication algorithm: md5 • Encryption algorithm: 3des-cbc

Table 74: IKE Phase 1 Configuration Parameters for the Initiator *(Continued)*

Feature	Name	Configuration Parameters
Policy	ike_pol	• Mode: main • Proposal reference: ike_prop • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	gate	• IKE policy reference: ike_pol • External interface: ge-0/0/1.0 • Gateway address: 1.1.100.23 • Local peer is hostname chicago • Remote peer is hostname sunnyvale

Table 75: IPsec Phase 2 Configuration Parameters for the Initiator

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-md5-96 • Encryption algorithm: 3des-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop • Perfect forward secrecy (PFS): group1
VPN	first_vpn	• IKE gateway reference: gate • IPsec policy reference: ipsec_pol

Table 76: Security Policy Configuration Parameters for the Initiator

Purpose	Name	Configuration Parameters
The security policy permits tunnel traffic from the trust zone to the untrust zone.	pol1	• Match criteria: • source-address any • destination-address any • application any • Action: permit tunnel ipsec-vpn first_vpn
The security policy permits tunnel traffic from the untrust zone to the trust zone.	pol1	• Match criteria: • application any • Action: permit tunnel ipsec-vpn first_vpn

See Table 77 on page 635 through Table 80 on page 638 for specific configuration parameters used for the responder in the examples.

Table 77: Interface, Routing Options, and Security Zones for the Responder

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0	13.168.11.100/24
	ge-0/0/1	10.2.99.1/24
Static routes	10.1.99.0/24 (default route)	The next hop is 13.168.11.100
	1.1.100.0/24	13.168.11.100

Table 77: Interface, Routing Options, and Security Zones for the Responder *(Continued)*

Feature	Name	Configuration Parameters
Security zones	trust	- All system services are allowed. - All protocols are allowed. - The ge-0/0/1.0 interface is bound to this zone.
	untrust	- The ge-0/0/0.0 interface is bound to this zone.

Table 78: IKE Phase 1 Configuration Parameters for the Responder

Feature	Name	Configuration Parameters
Proposal	ike_prop	- Authentication method: pre-shared-keys - Diffie-Hellman group: group2 - Authentication algorithm: md5 - Encryption algorithm: 3des-cbc
Policy	ike_pol	- Mode: main - Proposal reference: ike_prop - IKE Phase 1 policy authentication method: pre-shared-key ascii-text

Table 78: IKE Phase 1 Configuration Parameters for the Responder *(Continued)*

Feature	Name	Configuration Parameters
Gateway	gate	• IKE policy reference: ike_pol • External interface: ge-0/0/1.0 • Gateway address: 1.1.100.22 • Always send dead-peer detection • Local peer is hostname sunnyvale • Remote peer is hostname chicago

Table 79: IPsec Phase 2 Configuration Parameters for the Responder

Feature	Name	Configuration Parameters
Proposal	ipsec_prop	• Protocol: esp • Authentication algorithm: hmac-md5-96 • Encryption algorithm: 3des-cbc
Policy	ipsec_pol	• Proposal reference: ipsec_prop • Perfect forward secrecy (PFS): group1
VPN	first_vpn	• IKE gateway reference: gate • IPsec policy reference: ipsec_pol

Table 80: Security Policy Configuration Parameters for the Responder

Purpose	Name	Configuration Parameters
The security policy permits tunnel traffic from the trust zone to the untrust zone.	pol1	• Match criteria: • source-address any • destination-address any • application any • Action: permit tunnel ipsec-vpn first_vpn
The security policy permits tunnel traffic from the untrust zone to the trust zone.	pol1	• Match criteria: • application any • Action: permit tunnel ipsec-vpn first_vpn

Configuration

IN THIS SECTION

Configuring Interface, Routing Options, and Security Zones for the Initiator

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 12.168.99.100/24
set interfaces ge-0/0/1 unit 0 family inet address 10.1.99.1/24
set routing-options static route 10.2.99.0/24 next-hop 12.168.99.1
set routing-options static route 1.1.100.0/24 next-hop 12.168.99.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure interfaces, static routes, and security zones:

1. Configure Ethernet interface information.

   ```
   [edit]
   user@host# set interfaces ge-0/0/0 unit 0 family inet address 12.168.99.100/24
   user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.1.99.1/24
   ```

2. Configure static route information.

   ```
   [edit]
   user@host# set routing-options static route 10.2.99.0/24 next-hop 12.168.99.1
   user@host# set routing-options static route 1.1.100.0/24 next-hop 12.168.99.1
   ```

3. Configure the trust security zone.

```
[edit ]
user@host# set security zones security-zone trust host-inbound-traffic protocols all
```

4. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/1.0
```

5. Specify system services for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```

6. Assign an interface to the untrust security zone.

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/0.0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security zones` commands If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 12.168.99.100/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.1.99.1/24;
            }
        }
    }
```

```
[edit]
user@host# show routing-options
    static {
        route 10.2.99.0/24 next-hop 12.168.99.1;
        route 1.1.100.0/24 next-hop 12.168.99.1;
    }
```

```
[edit]
user@host# show security zones
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE for the Initiator

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike_prop authentication-method pre-shared-keys
set security ike proposal ike_prop dh-group group2
set security ike proposal ike_prop authentication-algorithm md5
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_pol mode aggressive
set security ike policy ike_pol proposals ike_prop
set security ike policy ike_pol pre-shared-key ascii-text "$ABC123”
set security ike gateway gate ike-policy ike_pol
set security ike gateway gate address 13.168.11.100
set security ike gateway gate external-interface ge-0/0/0.0
set security ike gateway gate local-identity hostname chicago
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE:

1. Create the IKE Phase 1 proposal.

```
[edit security ike]
user@host# edit proposal ike_prop
```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal ike_prop]
user@host# set authentication-method pre-shared-keys
```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike_prop]
user@host# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike_prop]
user@host# set authentication-algorithm md5
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike_prop]
user@host# set encryption-algorithm 3des-cbc
```

6. Create an IKE Phase 1 policy.

```
[edit security ike policy ]
user@host# edit policy ike_pol
```

7. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike_pol]
user@host# set mode aggressive
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ike_pol]
user@host# set proposals ike_prop
```

9. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike_pol pre-shared-key]
user@host# set ascii-text "$ABC123”
```

10. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike ]
user@host# set gateway gate external-interface ge-0/0/0.0
```

11. Create an IKE Phase 1 gateway address.

```
[edit security ike gateway gate]
set address 13.168.11.100
```

12. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway gate]
set ike-policy ike_pol
```

13. Set `local-identity` for the local peer.

```
[edit security ike gateway gate]
user@host# set local-identity hostname chicago
```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show security ike
    proposal ike_prop {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm md5;
        encryption-algorithm 3des-cbc;
    }
    policy ike_pol {
        mode aggressive;
        proposals ike_prop;
        pre-shared-key ascii-text "$ABC123”
    }
    gateway gate {
        ike-policy ike_pol;
        address 13.168.11.100;
        local-identity hostname chicago;
        external-interface ge-0/0/0.0;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec for the Initiator

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec policy ipsec_pol perfect-forward-secrecy keys group1
set security ipsec policy ipsec_pol proposals ipsec_prop
set security ipsec vpn first_vpn ike gateway gate
set security ipsec vpn first_vpn ike ipsec-policy ipsec_pol
set security ipsec vpn first_vpn establish-tunnels immediately
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

   ```
   [edit]
   user@host# edit security ipsec proposal ipsec_prop
   ```

2. Specify the IPsec Phase 2 proposal protocol.

```
[edit security ipsec proposal ipsec_prop]
user@host# set protocol esp
```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set authentication-algorithm hmac-md5-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set encryption-algorithm 3des-cbc
```

5. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec_pol]
user@host# set proposals ipsec_prop
```

6. Specify IPsec Phase 2 to use perfect forward secrecy (PFS) group1.

```
[edit security ipsec policy ipsec_pol ]
user@host# set perfect-forward-secrecy keys group1
```

7. Specify the IKE gateway.

```
[edit security ipsec]
user@host# set vpn first_vpn ike gateway gate
```

8. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn first_vpn ike ipsec-policy ipsec_pol
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show security ipsec
    proposal ipsec_prop {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm 3des-cbc;
    }
    policy ipsec_pol {
        perfect-forward-secrecy {
            keys group1;
        }
        proposals ipsec_prop;
    }
    vpn first_vpn {
        ike {
            gateway gate;
            ipsec-policy ipsec_pol;
        }
        establish-tunnels immediately;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Security Policies for the Initiator

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security policies from-zone trust to-zone untrust policy pol1 match source-address any
set security policies from-zone trust to-zone untrust policy pol1 match destination-address any
set security policies from-zone trust to-zone untrust policy pol1 match application any
set security policies from-zone trust to-zone untrust policy pol1 then permit tunnel ipsec-vpn
```

```
first_vpn
set security policies from-zone untrust to-zone trust policy pol1 match application any
set security policies from-zone untrust to-zone trust policy pol1 then permit tunnel ipsec-vpn
first_vpn
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure security policies:

1. Create the security policy to permit traffic from the trust zone to the untrust zone.

   ```
   [edit security policies from-zone trust to-zone untrust]
   user@host# set policy pol1 match source-address any
   user@host# set policy pol1 match destination-address any
   user@host# set policy pol1 match application any
   user@host# set policy pol1 then permit tunnel ipsec-vpn first_vpn
   ```

2. Create the security policy to permit traffic from the untrust zone to the trust zone.

   ```
   [edit security policies from-zone untrust to-zone trust]
   user@host# set policy pol1 match application any
   user@host# set policy pol1 then permit tunnel ipsec-vpn first_vpn
   ```

Results

From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show security policies
    from-zone trust to-zone untrust {
        policy pol1 {
            match {
                source-address any;
                destination-address any;
```

```
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-vpn first_vpn;
                    }
                }
            }
        }
    }
    from-zone untrust to-zone trust {
        policy pol1 {
            match {
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-vpn first_vpn;
                    }
                }
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring NAT for the Initiator

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security nat source rule-set ipsec from zone trust
set security nat source rule-set ipsec to zone untrust
set security nat source rule-set ipsec rule 1 match source-address 0.0.0.0/0
set security nat source rule-set ipsec rule 1 then source-nat interface
set security policies from-zone trust to-zone untrust policy allow-all match source-address any
set security policies from-zone trust to-zone untrust policy allow-all match destination-address
```

```
any
set security policies from-zone trust to-zone untrust policy allow-all match application any
set security policies from-zone trust to-zone untrust policy allow-all then permit
set security policies from-zone untrust to-zone trust policy allow-all match application any
set security policies from-zone untrust to-zone trust policy allow-all then permit
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet address 12.168.99.1/24
set interfaces ge-0/0/1 unit 0 family inet address 1.1.100.23/24
set routing-options static route 0.0.0.0/0 next-hop 1.1.100.22
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the initiator providing NAT:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet address 12.168.99.1/24
user@host# set ge-0/0/1 unit 0 family inet address 1.1.100.23/24
```

2. Configure zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/0.0
```

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/1.0
```

3. Configure NAT.

```
[edit security nat source rule-set ipsec]
user@host# set from zone trust
user@host# set to zone untrust
user@host# set rule 1 match source-address 0.0.0.0/0
user@host# set rule 1 then source-nat interface
```

4. Configure the default security policy.

```
[edit security policies]
user@host# set from-zone trust to-zone untrust policy allow-all match source-address any
user@host# set from-zone trust to-zone untrust policy allow-all match destination-address any
user@host# set from-zone trust to-zone untrust policy allow-all match application any
user@host# set from-zone trust to-zone untrust policy allow-all then permit
user@host# set from-zone untrust to-zone trust policy allow-all match application any
user@host# set from-zone untrust to-zone trust policy allow-all then permit
```

5. Configure the routing option.

```
[edit routing-options
user@host# set static route 0.0.0.0/0 next-hop 1.1.100.22
```

Results

From configuration mode, confirm your configuration by entering the `show security nat` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show security nat
        source {
            rule-set ipsec {
                from zone trust;
                to zone untrust;
                rule 1 {
                    match {
                        source-address 0.0.0.0/0;
                    }
                    then {
                        source-nat {
                            interface;
                        }
                    }
                }
            }
        }
    }
    policies {
        from-zone trust to-zone untrust {
            policy allow-all {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        from-zone untrust to-zone trust {
            policy allow-all {
                match {
                    application any;
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone trust {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
```

```
            interfaces {
                ge-0/0/0.0;
            }
        }
        security-zone untrust {
            host-inbound-traffic {
            }
            interfaces {
                ge-0/0/1.0;
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 12.168.99.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 1.1.100.23/24;
            }
        }
    }
}
routing-options {
    static {
        route 0.0.0.0/0 next-hop 1.1.100.22;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Interface, Routing Options, and Security Zones for the Responder

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 13.168.11.100/24
set interfaces ge-0/0/1 unit 0 family inet address 10.2.99.1/24
set routing-options static route 10.1.99.0/24 next-hop 13.168.11.1
set routing-options static route 1.1.100.0/24 next-hop 13.168.11.1
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure interfaces, static routes, security zones, and security policies:

1. Configure Ethernet interface information.

   ```
   [edit]
   user@host# set interfaces ge-0/0/0 unit 0 family inet address 13.168.11.100/24
   user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.2.99.1/24
   ```

2. Configure static route information.

   ```
   [edit]
   user@host# set routing-options static route 10.1.99.0/24 next-hop 13.168.11.1
   user@host# set routing-options static route 1.1.100.0/24 next-hop 13.168.11.1
   ```

3. Assign an interface to the untrust security zone.

```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/0.0
```

4. Configure the trust security zone.

```
[edit]
user@host# set security zones security-zone trust host-inbound-traffic protocols all
```

5. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/1.0
```

6. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security zones` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 13.168.11.100/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.2.99.1/24;
            }
        }
    }
```

```
[edit]
user@host# show routing-options
    static {
        route 10.1.99.0/24 next-hop 13.168.11.1;
        route 1.1.100.0/24 next-hop 13.168.11.1;
    }
```

```
[edit]
user@host# show security zones
    security-zone untrust {
        host-inbound-traffic {
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IKE for the Responder

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ike proposal ike_prop authentication-method pre-shared-keys
set security ike proposal ike_prop dh-group group2
set security ike proposal ike_prop authentication-algorithm md5
set security ike proposal ike_prop encryption-algorithm 3des-cbc
set security ike policy ike_pol mode aggressive
set security ike policy ike_pol proposals ike_prop
set security ike policy ike_pol pre-shared-key ascii-text "$ABC123"
set security ike gateway gate ike-policy ike_pol
set security ike gateway gate dynamic hostname chicago
set security ike gateway gate external-interface ge-0/0/0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IKE:

1. Define the IKE proposal authentication method.

   ```
   [edit security ike proposal ike_prop]
   user@host# set authentication-method pre-shared-key
   ```

2. Define the IKE proposal Diffie-Hellman group.

   ```
   [edit security ike proposal ike_prop]
   user@host# set dh-group group2
   ```

3. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike_prop]
user@host# set authentication-algorithm md5
```

4. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike_prop]
user@host# set encryption-algorithm 3des-cbc
```

5. Create an IKE Phase 1 policy.

```
[edit security ike]
user@host# edit policy ike_pol
```

6. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike_pol]
user@host# set mode aggressive
```

7. Specify a reference to the IKE proposal.

```
[edit security ike policy ike_pol]
user@host# set proposals ike_prop
```

8. Define the IKE Phase 1 policy authentication method.

```
[edit security ike policy ike_pol]
user@host# set pre-shared-key ascii-text "$ABC123"
```

9. Create an IKE Phase 1 gateway and define its dynamic host name.

```
[edit security ike gateway gate]
user@host# set dynamic hostname chicago
```

10. Create an IKE Phase 1 gateway and define its external interface.

    ```
    [edit security ike gateway gate]
    user@host# set external-interface ge-0/0/0.0
    ```

11. Define the IKE Phase 1 policy reference.

    ```
    [edit security ike gateway gate]
    user@host# set ike-policy ike_pol
    ```

Results

From configuration mode, confirm your configuration by entering the `show security ike` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show security ike
    proposal ike_prop {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm md5;
        encryption-algorithm 3des-cbc;
    }
    policy ike_pol {
        mode aggressive;
        proposals ike_prop;
        pre-shared-key ascii-text "$ABC123";
    }
    gateway gate {
        ike-policy ike_pol;
        dynamic hostname chicago;
        external-interface ge-0/0/0.0;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring IPsec for the Responder

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security ipsec proposal ipsec_prop protocol esp
set security ipsec proposal ipsec_prop authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
set security ipsec policy ipsec_pol perfect-forward-secrecy keys group1
set security ipsec policy ipsec_pol proposals ipsec_prop
set security ipsec vpn first_vpn ike gateway gate
set security ipsec vpn first_vpn ike ipsec-policy ipsec_pol
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

   ```
   [edit]
   user@host# edit security ipsec proposal ipsec_prop
   ```

2. Specify the IPsec Phase 2 proposal protocol.

   ```
   [edit security security ipsec proposal ipsec_prop]
   user@host# set protocol esp
   ```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

   ```
   [edit security ipsec proposal ipsec_prop]
   user@host# set authentication-algorithm hmac-md5-96
   ```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec_prop]
user@host# set encryption-algorithm 3des-cbc
```

5. Create the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# edit policy ipsec_pol
```

6. Set IPsec Phase 2 to use perfect forward secrecy (PFS) group1.

```
[edit security ipsec policy ipsec_pol]
user@host# set perfect-forward-secrecy keys group1
```

7. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec_pol]
user@host# set proposals ipsec_prop
```

8. Specify the IKE gateway.

```
[edit security ipsec]
user@host# set vpn first_vpn ike gateway gate
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@host# set vpn first_vpn ike ipsec-policy ipsec_pol
```

Results

From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show security ipsec
    proposal ipsec_prop {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm 3des-cbc;
    }
    policy ipsec_pol {
        perfect-forward-secrecy {
            keys group1;
        }
        proposals ipsec_prop;
    }
    vpn first_vpn {
        ike {
            gateway gate;
            ipsec-policy ipsec_pol;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Security Policies for the Responder

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security policies from-zone trust to-zone untrust policy pol1 match source-address any
set security policies from-zone trust to-zone untrust policy pol1 match destination-address any
set security policies from-zone trust to-zone untrust policy pol1 match application any
set security policies from-zone trust to-zone untrust policy pol1 then permit tunnel ipsec-vpn
first_vpn
```

```
set security policies from-zone untrust to-zone trust policy pol1 match application any
set security policies from-zone untrust to-zone trust policy pol1 then permit tunnel ipsec-vpn
first_vpn
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure security policies:

1. Create the security policy to permit traffic from the trust zone to the untrust zone.

   ```
   [edit security policies from-zone trust to-zone untrust]
   user@host# set policy pol1 match source-address any
   user@host# set policy pol1 match destination-address any
   user@host# set policy pol1 match application any
   user@host# set policy pol1 then permit tunnel ipsec-vpn first_vpn
   ```

2. Create the security policy to permit traffic from the untrust zone to the trust zone.

   ```
   [edit security policies from-zone untrust to-zone trust]
   user@host# set policy pol1 match application any
   user@host# set policy pol1 then permit tunnel ipsec-vpn first_vpn
   ```

Results

From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show security policies
    from-zone trust to-zone untrust {
        policy pol1 {
            match {
                source-address any;
                destination-address any;
                application any;
```

```
            }
            then {
                permit {
                    tunnel {
                        ipsec-vpn first_vpn;
                    }
                }
            }
        }
    }
    from-zone untrust to-zone trust {
        policy pol1 {
            match {
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-vpn first_vpn;
                    }
                }
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring NAT for the Responder

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security nat source rule-set ipsec from zone trust
set security nat source rule-set ipsec to zone untrust
set security nat source rule-set ipsec rule 1 match source-address 0.0.0.0/0
set security nat source rule-set ipsec rule 1 then source-nat interface
set security policies from-zone trust to-zone untrust policy allow-all match source-address any
set security policies from-zone trust to-zone untrust policy allow-all match destination-address
any
```

```
set security policies from-zone trust to-zone untrust policy allow-all match application any
set security policies from-zone trust to-zone untrust policy allow-all then permit
set security policies from-zone untrust to-zone trust policy allow-all match application any
set security policies from-zone untrust to-zone trust policy allow-all then permit
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet address 13.168.11.1/24
set interfaces ge-0/0/1 unit 0 family inet address 1.1.100.22/24
set routing-options static route 0.0.0.0/0 next-hop 1.1.100.23
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the responder providing NAT:

1. Configure interfaces.

   ```
   [edit interfaces]
   user@host# set ge-0/0/0 unit 0 family inet address 13.168.11.1/24
   user@host# set ge-0/0/1 unit 0 family inet address 1.1.100.22/24
   ```

2. Configure zones.

   ```
   [edit security zones security-zone trust]
   user@host# set host-inbound-traffic system-services all
   user@host# set host-inbound-traffic protocols all
   user@host# set interfaces ge-0/0/0.0
   ```

   ```
   [edit security zones security-zone untrust]
   user@host# set interfaces ge-0/0/1.0
   ```

3. Configure NAT.

   ```
   [edit security nat source rule-set ipsec]
   user@host# set from zone trust
   ```

```
user@host# set to zone untrust
user@host# set rule 1 match source-address 0.0.0.0/0
user@host# set rule 1 then source-nat interface
```

4. Configure the default security policy.

```
[edit security policies]
user@host# set from-zone trust to-zone untrust policy allow-all match source-address any
user@host# set from-zone trust to-zone untrust policy allow-all match destination-address any
user@host# set from-zone trust to-zone untrust policy allow-all match application any
user@host# set from-zone trust to-zone untrust policy allow-all then permit
user@host# set from-zone untrust to-zone trust policy allow-all match application any
user@host# set from-zone untrust to-zone trust policy allow-all then permit
```

5. Configure the routing option.

```
[edit routing-options
user@host# set static route 0.0.0.0/0 next-hop 1.1.100.23
```

Results

From configuration mode, confirm your configuration by entering the `show security nat` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show security nat
    nat {
        source {
            rule-set ipsec {
                from zone trust;
                to zone untrust;
                rule 1 {
                    match {
                        source-address 0.0.0.0/0;
                    }
                    then {
                        source-nat {
                            interface;
                        }
                    }
                }
            }
        }
    }
    policies {
        from-zone trust to-zone untrust {
            policy allow-all {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        from-zone untrust to-zone trust {
            policy allow-all {
                match {
                    application any;
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone trust {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                ge-0/0/0.0;
            }
        }
        security-zone untrust {
            host-inbound-traffic {
            }
            interfaces {
                ge-0/0/1.0;
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 13.168.11.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 1.1.100.22/24;
            }
        }
    }
}
routing-options {
    static {
        route 0.0.0.0/0 next-hop 1.1.100.23;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the IKE Phase 1 Status for the Initiator

Purpose

Verify the IKE Phase 1 status.

Action

Before starting the verification process, you must send traffic from a host in the 10.1.99.0 network to a host in the 10.2.99.0 network. For route-based VPNs, traffic can be initiated by the SRX Series Firewall through the tunnel. We recommend that when testing IPsec tunnels, test traffic be sent from a separate device on one side of the VPN to a second device on the other side of the VPN. For example, initiate a ping operation from 10.1.99.2 to 10.2.99.2.

From operational mode, enter the `show security ike security-associations` command. After obtaining an index number from the command, use the `show security ike security-associations index` *`index_number`* `detail` command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
5649304 UP     c3193077d38e426f  011f0ef28d928f4c  Aggressive     13.168.11.
```

```
user@host> show security ike security-associations index 5649304 detail
IKE peer 13.168.11.100, Index 5649304, Gateway Name: gate
  Role: Initiator, State: UP
  Initiator cookie: c3193077d38e426f, Responder cookie: 011f0ef28d928f4c
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 12.168.99.100:4500, Remote: 13.168.11.100:4500
  Lifetime: Expires in 26359 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
  Remote Access Client Info: Unknown Client
  Peer ike-id: 13.168.11.100
```

```
    AAA assigned IP: 0.0.0.0
    Algorithms:
     Authentication        : hmac-md5-96
     Encryption            : 3des-cbc
     Pseudo random function: hmac-md5
     Diffie-Hellman group  : DH-group-2
    Traffic statistics:
     Input  bytes  :                 1140
     Output bytes  :                 1203
     Input  packets:                    6
     Output packets:                    6
     Input  fragmentated packets:       0
     Output fragmentated packets:       0
    IPSec security associations: 2 created, 3 deleted
    Phase 2 negotiations in progress: 1

      Negotiation type: Quick mode, Role: Initiator, Message ID: 0
      Local: 12.168.99.100:4500, Remote: 13.168.11.100:4500
      Local identity: chicago
      Remote identity: 13.168.11.100
      Flags: IKE SA is created
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.
- Remote address—Verify that the remote IP address is correct and that port 4500 is being used for peer-to-peer communication.
- Role initiator state
  - Up—The Phase 1 SA has been established.
  - Down—There was a problem establishing the Phase 1 SA.
  - Both peers in the IPsec SA pair are using port 4500, which indicates that NAT-T is implemented. (NAT-T uses port 4500 or another random high-numbered port.)

  - Peer IKE ID—Verify the remote (responder) ID is correct. In this example, the hostname is sunnyvale.

  - Local identity and remote identity—Verify these are correct.

- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)

- IKE policy parameters

- Preshared key information

- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations` command lists additional information about security associations:

- Authentication and encryption algorithms used

- Phase 1 lifetime

- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)

- Role information

  Troubleshooting is best performed on the peer using the responder role.

- Initiator and responder information

- Number of IPsec SAs created

- Number of Phase 2 negotiations in progress

Verifying IPsec Security Associations for the Initiator

Purpose

Verify the IPsec status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *`index_number`* `detail` command.

```
user@host> show security ipsec security-associations

Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <2      ESP:3des/md5    aff3ac30 1103/ unlim  -   root 4500  13.168.11.100
  >2      ESP:3des/md5    40539d12 1103/ unlim  -   root 4500  13.168.11.100
```

```
user@host> show security ipsec security-associations detail

ID: 2 Virtual-system: root, VPN Name: first_vpn
  Local Gateway: 12.168.99.100, Remote Gateway: 13.168.11.100
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Disabled                             , Policy-name: pol1
  Port: 4500, Nego#: 7, Fail#: 0, Def-Del#: 0 Flag: 0x600829
  Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
  Tunnel events:
    Wed Apr 08 2020 19:13:53: IPSec SA negotiation successfully completed (1 times)
    Wed Apr 08 2020
    : IPSec SA delete payload received from peer, corresponding IPSec SAs cleared (1 times)
    Wed Apr 08 2020 19:13:09: IPSec SA negotiation successfully completed (1 times)
    Wed Apr 08 2020 19:13:09: User cleared IPSec SA from CLI (1 times)
    Wed Apr 08 2020 19:13:09: IKE SA negotiation successfully completed (5 times)
    Wed Apr 08 2020 19:12:18: IPSec SA negotiation successfully completed (1 times)
    Wed Apr 08 2020 19:12:18: User cleared IPSec SA from CLI (1 times)
    Wed Apr 08 2020 19:12:12: IPSec SA negotiation successfully completed (1 times)
    Wed Apr 08 2020 19:12:12: User cleared IPSec SA from CLI (1 times)
    Wed Apr 08 2020 19:06:52: Peer's IKE-ID validation failed during negotiation (2 times)
    Wed Apr 08 2020
    : Negotiation failed  with error code NO_PROPOSAL_CHOSEN received from peer (2 times)
    Wed Apr 08 2020 19:05:26: Peer's IKE-ID validation failed during negotiation (1 times)
    Wed Apr 08 2020
    : Negotiation failed  with error code NO_PROPOSAL_CHOSEN received from peer (1 times)
    Wed Apr 08 2020 19:04:26: Peer's IKE-ID validation failed during negotiation (1 times)
    Wed Apr 08 2020
    : Negotiation failed  with error code NO_PROPOSAL_CHOSEN received from peer (1 times)
    Wed Apr 08 2020 19:03:26: Peer's IKE-ID validation failed during negotiation (1 times)
  Direction: inbound, SPI: aff3ac30, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1093 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 453 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-md5-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 40539d12, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1093 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 453 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-md5-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The remote gateway has a NAT address of 13.168.11.100.
- Both peers in the IPsec SA pair are using port 4500, which indicates that NAT-T is implemented. (NAT-T uses port 4500 or another random high-numbered port.).
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3390/ unlimited value indicates that the Phase 2 lifetime expires in 3390 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.
- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

Verifying the IKE Phase 1 Status for the Responder

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the `show security ike security-associations` command. After obtaining an index number from the command, use the `show security ike security-associations index` *`index_number`* `detail` command.

```
user@host> show security ike security-associations

Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
2914355 UP     c3193077d38e426f  011f0ef28d928f4c  Aggressive     1.1.100.23
```

```
user@host> show security ike security-associations index 2914355 detail

  IKE peer 1.1.100.23, Index 2914355, Gateway Name: gate
  Role: Responder, State: UP
  Initiator cookie: c3193077d38e426f, Responder cookie: 011f0ef28d928f4c
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 13.168.11.100:4500, Remote: 1.1.100.23:23434
  Lifetime: Expires in 26137 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Disabled, Size: 0
  Remote Access Client Info: Unknown Client
  Peer ike-id: chicago
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-md5-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-md5
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :                1203
   Output bytes  :                1140
   Input  packets:                   6
   Output packets:                   6
   Input  fragmented packets:      0
   Output fragmented packets:      0
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 13.168.11.100:4500, Remote: 1.1.100.23:23434
    Local identity: 13.168.11.100
    Remote identity: chicago
    Flags: IKE SA is created
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.

- Remote address—Verify that the remote IP address is correct and that port 4500 is being used for peer-to-peer communication.

- Role responder state

  - Up—The Phase 1 SA has been established.

  - Down—There was a problem establishing the Phase 1 SA.

  - Peer IKE ID—Verify the local ID for the peer is correct. In this example, the hostname is chicago.

  - Local identity and remote identity—Verify these are correct.

- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)

- IKE policy parameters

- Preshared key information

- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations` command lists additional information about security associations:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information

  Troubleshooting is best performed on the peer using the responder role.
- Initiator and responder information
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

Verifying IPsec Security Associations for the Responder

Purpose

Verify the IPsec status.

Action

From operational mode, enter the `show security ipsec security-associations` command. After obtaining an index number from the command, use the `show security ipsec security-associations index` *`index_number`* `detail` command.

```
user@host> show security ipsec security-associations

Total active tunnels: 1     Total Ipsec sas: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108878 ESP:3des/md5  40539d12 939/ unlim   -   root 23434 1.1.100.23
  >67108878 ESP:3des/md5  aff3ac30 939/ unlim   -   root 23434 1.1.100.23
```

```
user@host> show security ipsec security-associations detail

  ID: 67108878 Virtual-system: root, VPN Name: first_vpn
  Local Gateway: 13.168.11.100, Remote Gateway: 1.1.100.23
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
```

```
    Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    Version: IKEv1
    DF-bit: clear, Copy-Outer-DSCP Disabled                             , Policy-name: pol1
    Port: 23434, Nego#: 8, Fail#: 0, Def-Del#: 0 Flag: 0x608829
    Multi-sa, Configured SAs# 1, Negotiated SAs#: 1
    Tunnel events:
      Wed Apr 08 2020 19:14:22: IPSec SA negotiation successfully completed (1 times)
      Wed Apr 08 2020 19:14:15: User cleared IPSec SA from CLI (1 times)
      Wed Apr 08 2020 19:13:39: IPSec SA negotiation successfully completed (3 times)
      Wed Apr 08 2020 19:13:39: IKE SA negotiation successfully completed (4 times)
      Wed Apr 08 2020
      : IPSec SA delete payload received from peer, corresponding IPSec SAs cleared (1 times)
      Wed Apr 08 2020 19:10:39: IPSec SA negotiation successfully completed (1 times)
      Wed Apr 08 2020 19:10:20: User cleared IPSec SA from CLI (1 times)
      Wed Apr 08 2020 19:10:08: IPSec SA negotiation successfully completed (1 times)
      Wed Apr 08 2020
      : Tunnel is ready. Waiting for trigger event or peer to trigger negotiation (1 times)
    Direction: inbound, SPI: 40539d12, AUX-SPI: 0
                                , VPN Monitoring: -
      Hard lifetime: Expires in 930 seconds
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expires in 335 seconds
      Mode: Tunnel(0 0), Type: dynamic, State: installed
      Protocol: ESP, Authentication: hmac-md5-96, Encryption: 3des-cbc
      Anti-replay service: counter-based enabled, Replay window size: 64
    Direction: outbound, SPI: aff3ac30, AUX-SPI: 0
                                , VPN Monitoring: -
      Hard lifetime: Expires in 930 seconds
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expires in 335 seconds
      Mode: Tunnel(0 0), Type: dynamic, State: installed
      Protocol: ESP, Authentication: hmac-md5-96, Encryption: 3des-cbc
      Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The remote gateway has a NAT address of 1.1.100.23.
- Both peers in the IPsec SA pair are using port 4500, which indicates that NAT-T is implemented. (NAT-T uses port 4500 or another random high-numbered port.)

- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3571/ unlim value indicates that the Phase 2 lifetime expires in 3571 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.
- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

SEE ALSO

IPsec Overview | 20
Understanding Policy-Based IPsec VPNs | 267

Example: Configuring NAT-T with Dynamic Endpoint VPN

IN THIS SECTION

This example shows how to configure a route-based VPN where the IKEv2 initiator is a dynamic endpoint behind a NAT device.

Requirements

This example uses the following hardware and software components:

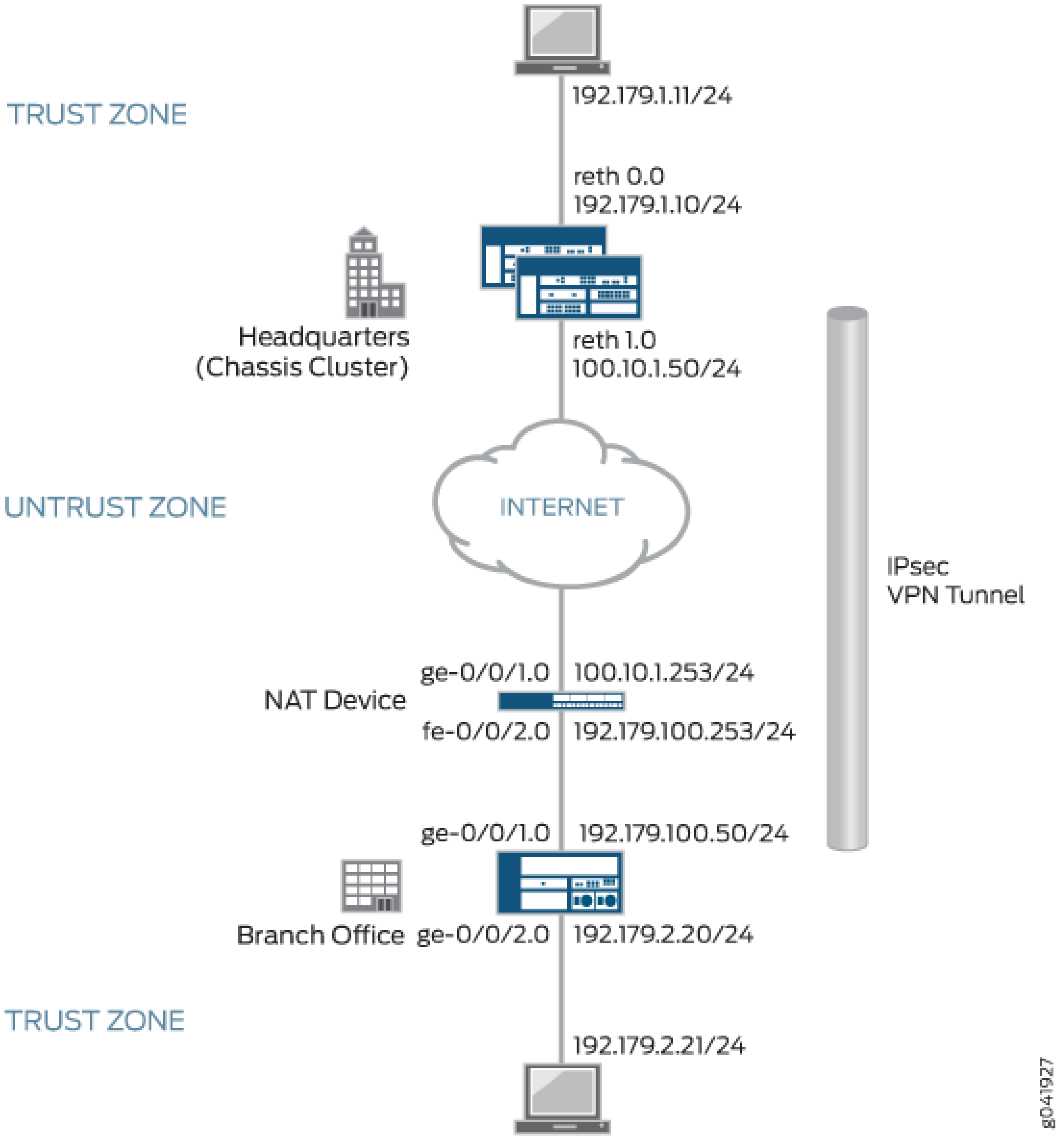
- Two SRX Series Firewalls configured in a chassis cluster
- One SRX Series Firewall providing NAT
- One SRX Series Firewall providing branch office network access
- Junos OS Release 12.1X46-D10 or later for IKEv2 NAT-T support

Overview

In this example, an IPsec VPN is configured between the branch office (IKEv2 initiator) and headquarters (IKEv2 responder) to secure network traffic between the two locations. The branch office is located behind the NAT device. The branch office address is assigned dynamically and is unknown to the responder. The initiator is configured with the remote identity of the responder for tunnel negotiation. This configuration establishes a dynamic endpoint VPN between the peers across the NAT device.

Figure 48 on page 680 shows an example of a topology with NAT-Traversal (NAT-T) and dynamic endpoint VPN.

Figure 48: NAT-T with Dynamic Endpoint VPN



In this example, the initiator's IP address, 192.179.100.50, which has been dynamically assigned to the device, is hidden by the NAT device and translated to 100.10.1.253.

The following configuration options apply in this example:

- The local identity configured on the initiator must match the remote gateway identity configured on the responder.
- Phase 1 and Phase 2 options must match between the initiator and responder.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Starting with Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, the default value for the `nat-keepalive` option configured at the `[edit security ike gateway gateway-name]` hierarchy level has been changed from 5 seconds to 20 seconds.

In SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices, IKE negotiations involving NAT traversal do not work if the IKE peer is behind a NAT device that will change the source IP address of the IKE packets during the negotiation. For example, if the NAT device is configured with DIP, it changes the source IP because the IKE protocol switches the UDP port from 500 to 4500. (Platform support depends on the Junos OS release in your installation.)

Configuration

IN THIS SECTION

Configuring the Branch Office Device (IKEv2 Initiator)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 192.179.100.50/24
set interfaces ge-0/0/2 unit 0 family inet address 192.179.2.20/24
set interfaces st0 unit 0 family inet address 172.168.100.1/16
set routing-options static route 192.179.1.0/24 next-hop st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/2.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway HQ_GW ike-policy IKE_POL
set security ike gateway HQ_GW address 100.10.1.50
set security ike gateway HQ_GW local-identity hostname branch.example.net
set security ike gateway HQ_GW external-interface ge-0/0/1.0
set security ike gateway HQ_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn HQ_VPN bind-interface st0.0
set security ipsec vpn HQ_VPN ike gateway HQ_GW
set security ipsec vpn HQ_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn HQ_VPN establish-tunnels immediately
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the branch office device:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 192.179.100.50/24
user@host# set ge-0/0/2 unit 0 family inet address 192.179.2.20/24
user@host# set st0 unit 0 family inet address 172.168.100.1/16
```

2. Configure routing options.

```
[edit routing-options]
user@host# set static route 192.179.1.0/24 next-hop st0.0
```

3. Configure zones.

```
[edit security zones security-zones trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/2.0
[edit security zones security-zones untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host#set interfaces st0.0
```

4. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
user@host# set proposals IKE_PROP
user@host# set pre-shared-key ascii-text "$ABC123"
 [edit security ike gateway HQ_GW]
user@host# set ike-policy IKE_POL
user@host# set address 100.10.1.50
user@host# set local-identity hostname branch.example.net
user@host# set external-interface ge-0/0/1.0
user@host# set version v2-only
```

5. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
```

```
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy IPSEC_POL]
user@host# set proposals IPSEC_PROP
user@host# set perfect-forward-secrecy keys group5
[edit security ipsec vpn HQ_VPN]
user@host# set bind-interface st0.0
user@host# set ike gateway HQ_GW
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

6. Configure the security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security zones`, `show security ike`, `show security ipsec`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.179.100.50/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.179.2.20/24;
        }
    }
}
st0 {
    unit 0 {
        family inet {
            address 172.168.100.1/16;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 192.179.1.0/24 next-hop st0.0;
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.0;
    }
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method pre-shared-keys;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    pre-shared-key ascii-text "$ABC123”
}
gateway HQ_GW{
    ike-policy IKE_POL;
    address 100.10.1.50;
    local-identity hostname branch.example.net;
    external-interface ge-0/0/1.0;
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-256-cbc;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals IPSEC_PROP;
}
vpn HQ_VPN {
    bind-interface st0.0;
    ike {
        gateway HQ_GW;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the NAT Device

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 100.10.1.253/24
set interfaces fe-0/0/2 unit 0 family inet address 192.179.100.253/24
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/2.0
set security nat source rule-set DYNAMIC from zone trust
set security nat source rule-set DYNAMIC to zone untrust
set security nat source rule-set DYNAMIC rule R2R3 match source-address 0.0.0.0/0
set security nat source rule-set DYNAMIC rule R2R3 then source-nat interface
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the intermediate router providing NAT:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 100.10.1.253/24
user@host# set fe-0/0/2 unit 0 family inet address 192.179.100.253/24
```

2. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
```

```
user@host# set interfaces ge-0/0/1.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/2.0
```

3. Configure NAT.

```
[edit security nat source rule-set DYNAMIC]
user@host# set from zone trust
user@host# set to zone untrust
user@host# set rule R2R3 match source-address 0.0.0.0/0
user@host# set rule R2R3 then source-nat interface
```

4. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security nat source`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 100.10.1.253/24;
        }
    }
}
fe-0/0/2 {
    unit 0 {
        family inet {
            address 192.179.100.253/24;
        }
    }
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-0/0/2.0;
    }
}
[edit]
user@host# show security nat source
rule-set DYNAMIC {
    from zone untrust;
    to zone trust;
    rule R2R3 {
        match {
            source-address 0.0.0.0/0;
        }
        then {
            source-nat {
                interface;
            }
        }
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Headquarters Device (IKEv2 Responder)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set chassis cluster reth-count 5
set chassis cluster redundancy-group 1 node 0 priority 220
set chassis cluster redundancy-group 1 node 1 priority 149
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-8/0/2 weight 255
set interfaces ge-0/0/1 gigether-options redundant-parent reth0
set interfaces ge-0/0/2 gigether-options redundant-parent reth1
set interfaces ge-8/0/1 gigether-options redundant-parent reth0
set interfaces ge-8/0/2 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 192.179.1.10/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 100.10.1.50/24
set interfaces st0 unit 0 family inet address 172.168.100.2/16
set routing-options static route 192.179.2.0/24 next-hop st0.0
set routing-options static route 192.179.100.0/24 next-hop 100.10.1.253
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces reth0.0
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway Branch_GW ike-policy IKE_POL
set security ike gateway Branch_GW dynamic hostname branch.example.net
set security ike gateway Branch_GW dead-peer-detection optimized
set security ike gateway Branch_GW external-interface reth1.0
set security ike gateway Branch_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn Branch_VPN bind-interface st0.0
set security ipsec vpn Branch_VPN ike gateway Branch_GW
set security ipsec vpn Branch_VPN ike ipsec-policy IPSEC_POL
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

1. Configure two nodes as the chassis cluster.

```
[edit chassis cluster]
user@host# set reth-count 5
user@host# set redundancy-group 1 node 0 priority 220
user@host# set redundancy-group 1 node 1 priority 149
user@host# set redundancy-group 1 interface-monitor ge-0/0/1 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/0/1 weight 255
user@host# set redundancy-group 1 interface-monitor ge-0/0/2 weight 255
user@host# set redundancy-group 1 interface-monitor ge-8/0/2 weight 255
```

2. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 gigether-options redundant-parent reth0
user@host# set ge-0/0/2 gigether-options redundant-parent reth1
user@host# set ge-8/0/1 gigether-options redundant-parent reth0
user@host# set ge-8/0/2 gigether-options redundant-parent reth1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet address 192.179.1.10/24
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 100.10.1.50/24
user@host# set st0 unit 0 family inet address 172.168.100.2/16
```

3. Configure routing options.

```
[edit routing-options]
user@host# set static route 192.179.2.0/24 next-hop st0.0
user@host# set static route 192.179.100.0/24 next-hop 100.10.1.253
```

4. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic protocols all
user@host# set host-inbound-traffic system-services all
user@host# set interfaces st0.0
user@host# set interfaces reth1.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces reth0.0
```

5. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
```

```
user@host# set proposals IKE_PROP
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security ike gateway Branch_GW]
user@host# set ike-policy IKE_POL
user@host# set dynamic hostname branch.example.net
user@host# set dead-peer-detection optimized
user@host# set external-interface reth1.0
user@host# set version v2-only
```

6. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn Branch_VPN]
user@host# set bind-interface st0.0
user@host# set ike gateway Branch_GW
user@host# set ike ipsec-policy IPSEC_POL
```

7. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show chassis cluster`, `show interfaces`, `show routing-options`, `show security zones`, `show security ike`, `show security ipsec`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis cluster
reth-count 5;
redundancy-group 1 {
    node 0 priority 220;
    node 1 priority 149;
    interface-monitor {
        ge-0/0/1 weight 255;
        ge-8/0/1 weight 255;
        ge-0/0/2 weight 255;
        ge-8/0/2 weight 255;
    }
}
[edit]
user@host# show interfaces
ge-0/0/1 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-0/0/2 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-8/0/1 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-8/0/2 {
    gigether-options {
        redundant-parent reth1;
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.179.1.10/24;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 100.10.1.50/24;
        }
    }
}
st0 {
    unit 0{
        family inet {
            address 172.168.100.2/16;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 192.179.2.0/24 next-hop st0.0;
    route 192.179.100.0/24 next-hop 100.10.1.253;
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        reth0.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.0;
        reth1.0;
    }
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method pre-shared-keys;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    pre-shared-key ascii-text “$ABC123”
}
gateway Branch_GW {
    ike-policy IKE_POL;

    dynamic hostname branch.example.net;
    dead-peer-detection optimized;
    external-interface reth1.0;
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-256-cbc;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals IPSEC_PROP;
}
vpn Branch_VPN {
    bind-interface st0.0;
    ike {
        gateway Branch_GW;
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the IKE Phase 1 Status for the Responder

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode on node 0, enter the **show security ike security-associations** command. After obtaining an index number from the command, use the **show security ike security-associations detail** command.

```
user@host# show security ike security-associations
node0:
Index      State  Initiator cookie   Responder cookie  Mode    Remote Address
1367024684 UP     f82c54347e2f3fb1   020e28e1e4cae003  IKEv2    100.10.1.253
```

```
user@host# show security ike security-associations detail
node0:
IKE peer 100.10.1.253, Index 1367024684, Gateway Name: Branch_GW
  Location: FPC 5, PIC 0, KMD-Instance 2
  Role: Responder, State: UP
  Initiator cookie: f82c54347e2f3fb1, Responder cookie: 020e28e1e4cae003
  Exchange type: IKEv2, Authentication method: Pre-shared-keys
  Local: 100.10.1.50:4500, Remote: 100.10.1.253:2541
  Lifetime: Expires in 3593 seconds
  Peer ike-id: branch.example.net
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                  683
   Output bytes  :                  400
   Input  packets:                    2
   Output packets:                    1
  IPSec security associations: 0 created, 0 deleted
  Phase 2 negotiations in progress: 1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index` *`index_id`* `detail` command to get more information about the SA.

- Remote address—Verify that the local IP address is correct and that port 4500 is being used for peer-to-peer communication.

- Role responder state

  - Up—The Phase 1 SA has been established.

  - Down—There was a problem establishing the Phase 1 SA.

  - Peer IKE ID—Verify the address is correct.

  - Local identity and remote identity—Verify these addresses are correct.

- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that sends IKE packets)

- IKE policy parameters

- Preshared key information

- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations` command lists additional information about security associations:

- Authentication and encryption algorithms used

- Phase 1 lifetime

- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)

- Role information

  Troubleshooting is best performed on the peer using the responder role.

- Initiator and responder information

- Number of IPsec SAs created

- Number of Phase 2 negotiations in progress

Verifying IPsec Security Associations for the Responder

Purpose

Verify the IPsec status.

Action

From operational mode on node 0, enter the **show security ipsec security-associations** command. After obtaining an index number from the command, use the **show security ipsec security-associations detail** command.

```
user@host# show security ipsec security-associations
node0
  Total active tunnels: 1
  ID        Algorithm           SPI      Life:sec/kb  Mon lsys Port Gateway
  <77856771 ESP:aes-cbc-256/sha1 4ad5af40 7186/unlim   - root   2541 100.10.1.253
  >77856771 ESP:aes-cbc-256/sha1 5bb0a5ee 7186/unlim   - root   2541 100.10.1.253
```

```
user@host# show security ipsec security-associations detail
node0
  ID: 77856771 Virtual-system: root, VPN Name: Branch_VPN
  Local Gateway: 100.10.1.50, Remote Gateway: 100.10.1.253
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
    DF-bit: clear
    Bind-interface: st0.0

  Port: 2541, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 608a29
  Tunnel Down Reason: SA not initiated
    Location: FPC 5, PIC 0, KMD-Instance 2
    Direction: inbound, SPI: 4ad5af40, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 7182 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 6587 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The output from the `show security ipsec security-associations` command lists the following information:

- The remote gateway has an IP address of 100.10.1.253.

- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The lifetime value indicates that the Phase 2 lifetime expires in 7186 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index` *`index_id`* `detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

  A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, match for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.
- Another common reason for Phase 2 failure is not specifying the ST interface binding. If IPsec cannot complete, check the kmd log or set trace options.

SEE ALSO

IPsec Overview | 20

Security Policies Overview

Release History Table

Release	Description
12.1X46-D10	Starting with Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, the default value for the `nat-keepalive` option configured at the `[edit security ike gateway` *`gateway-name`*`]` hierarchy level has been changed from 5 seconds to 20 seconds.

RELATED DOCUMENTATION

Traffic Selectors in Route-Based VPNs | 524

10 CHAPTER

Group VPN

Group VPNv1

IN THIS SECTION

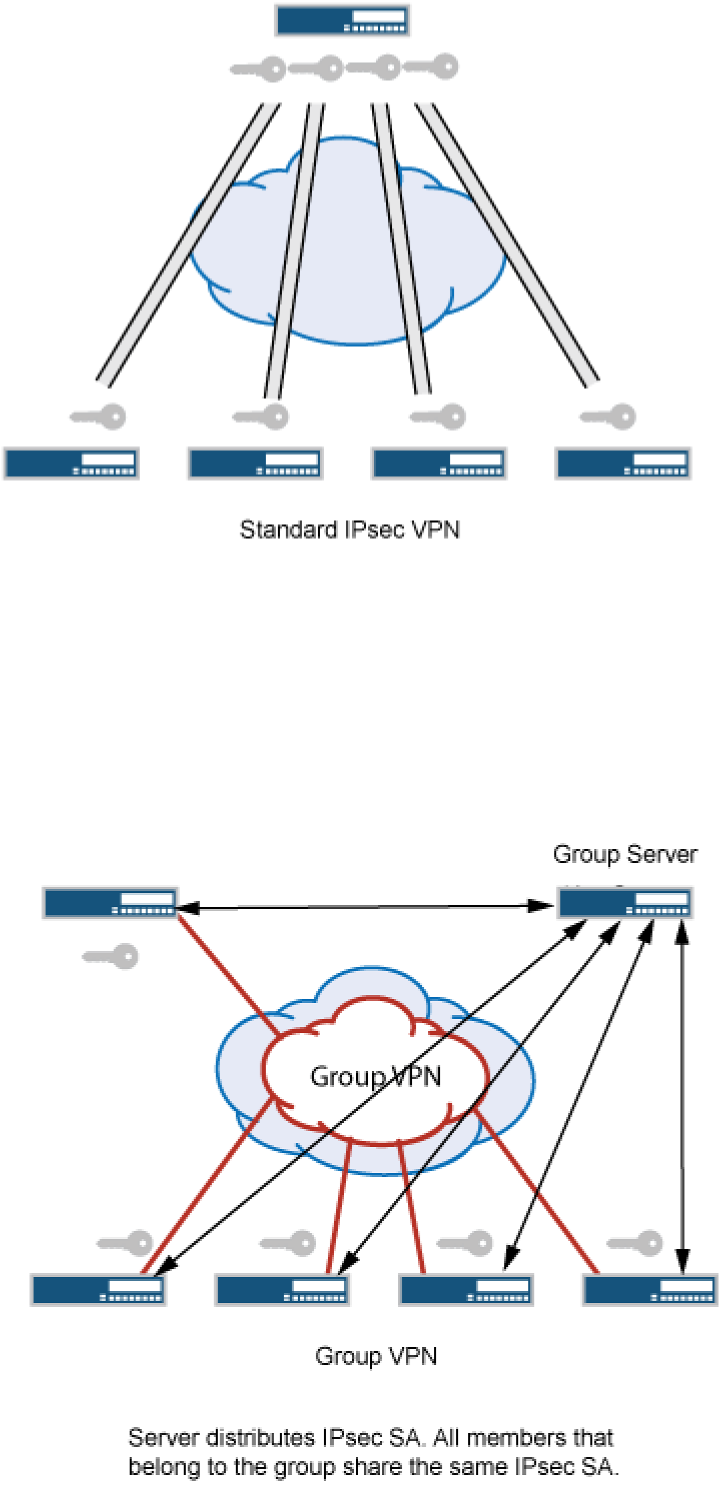
Group VPN is a set of features that are necessary to secure IP multicast group traffic or unicast traffic over a private WAN that originates on or flows through a device.

Group VPNv1 Overview

IN THIS SECTION

An IPsec security association (SA) is a unidirectional agreement between virtual private network (VPN) participants that defines the rules to use for authentication and encryption algorithms, key exchange mechanisms, and secure communications. With current VPN implementations, the SA is a point-to-point tunnel between two security devices. Group VPNv1 extends IPsec architecture to support SAs that are shared by a group of security devices (see Figure 49 on page 705).

Figure 49: Standard IPsec VPN and Group VPNv1


Standard IPsec VPN
Group Server
Group VPN
Group VPN
Server distributes IPsec SA. All members that belong to the group share the same IPsec SA.

Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices. With Group VPNv1, any-to-any connectivity is achieved by preserving the original source and destination IP addresses in the outer header. Secure multicast packets are replicated in the same way as cleartext multicast packets in the core network.

Starting with Junos OS Release 12.3X48-D30, Group VPNv1 members can interoperate with Group VPNv2 servers.

Group VPNv1 has some propriety limitations regarding RFC 6407, *The Group Domain of Interpretation (GDOI)*. To use Group VPN without proprietary limitations, upgrade to Group VPNv2. Group VPNv2 is supported on vSRX Virtual Firewall instances starting with Junos OS Release 15.1X49-D30, SRX Series Firewalls starting with Junos OS Release 15.1X49-D40, and MX Series devices starting with Junos OS Release 15.1r2.

Understanding the GDOI Protocol for Group VPNv1

Group VPNv1 is based on RFC 3547, *The Group Domain of Interpretation* (GDOI). This RFC describes the protocol between group members and a group server to establish SAs among group members. GDOI messages create, maintain, or delete SAs for a group of devices. The GDOI protocol runs on port 848.

The Internet Security Association and Key Management Protocol (ISAKMP) defines two negotiation phases to establish SAs for an AutoKey IKE IPsec tunnel. Phase 1 allows two devices to establish an ISAKMP SA. Phase 2 establishes SAs for other security protocols, such as GDOI.

With group VPN, Phase 1 ISAKMP SA negotiation is performed between a group server and a group member. The server and member must use the same ISAKMP policy. In Phase 2, GDOI exchanges between the server and member establish the SAs that are shared with other group members. A group member does not need to negotiate IPsec with other group members. GDOI exchanges in Phase 2 must be protected by ISAKMP Phase 1 SAs.

There are two types of GDOI exchanges:

- The `groupkey-pull` exchange allows a member to request SAs and keys shared by the group from the server.
- The `groupkey-push` exchange is a single rekey message that allows the server to send group SAs and keys to members before existing group SAs expire. Rekey messages are unsolicited messages sent from the server to members.

Understanding Group VPNv1 Limitations

The following are not supported in this release for group VPNv1:

- Non-default routing instances
- *Chassis cluster*
- Server clusters

- Route-based group VPN
- Public Internet-based deployment
- SNMP
- Deny policy from Cisco GET VPN server
- J-Web interface for configuration and monitoring

Starting with Junos OS Release 12.3X48-D30, Group VPNv1 members on SRX100, SRX110, SRX210, SRX220, SRX240, SRX550, and SRX650 devices can interoperate with Group VPNv2 servers. When you configure Group VPNv1 members for use with Group VPNv2 servers, note the following limitations:

- Group VPNv2 supports the IETF draft specification *IP Delivery Delay Detection Protocol* for a time-based antireplay mechanism. Therefore, IP delivery delay detection protocol-based antireplay is not supported on Group VPNv1 members and must be disabled on the Group VPNv2 server with the `deactivate security group-vpn server group group-name anti-replay-time-window` command.
- The Group VPNv2 server does not support colocation, where the group server and group member functions exist in the same device.
- The Group VPNv2 server does not support heartbeat transmittals. Heartbeat must be disabled on the Group VPNv1 member with the `deactivate security group-vpn member ipsec vpn vpn-name heartbeat-threshold` command. We recommend using Group VPNv2 server clusters to avoid traffic impact due to reboots or other interruptions on the Group VPNv2 server.
- Groupkey-push messages sent from the Group VPNv2 server are based on RFC 6407, *The Group Domain of Interpretation (GDOI)* and are not supported on Group VPNv1 members. Therefore, groupkey-push messages must be disabled on the Group VPNv2 server with the `deactivate security group-vpn server group group-name server-member-communication` command.

  Rekeys are supported with groupkey-pull messages. If there are scaling issues where Group VPNv1 members cannot complete the groupkey-pull operation before the TEK hard lifetime expires, we recommend increasing the TEK lifetime to allow sufficient time for members to complete the groupkey-pull operation. Juniper’s scaling numbers are qualified with a 2 hour TEK lifetime.
- If the Group VPNv2 server is rebooted or upgraded, or the SAs for the group are cleared, new members cannot be added to the network until the next rekey occurs for existing members. New members cannot send traffic to existing members that have old keys. As a workaround, clear the SAs on the existing Group VPNv1 members with the `clear security group-vpn member ipsec security-associations` command.
- Because multicast data traffic is not supported by Group VPNv2 members, multicast data traffic cannot be used when Group VPNv1 and Group VPNv2 members coexist in the network for the same group.

Understanding Group VPNv1 Servers and Members

The center of a group VPN is the group server. The group server performs the following tasks:

- Controls group membership
- Generates encryption keys
- Manages group SAs and keys and distributes them to group members

Group members encrypt traffic based on the group SAs and keys provided by the group server.

A group server can service multiple groups. A single security device can be a member of multiple groups.

Each group is represented by a group identifier, which is a number between 1 and 65,535. The group server and group members are linked together by the group identifier. There can be only one group identifier per group, and multiple groups cannot use the same group identifier.

The following is a high-level view of group VPN server and member actions:

1. The group server listens on UDP port 848 for members to register. A member device must provide correct IKE Phase 1 authentication to join the group. Preshared key authentication on a per-member basis is supported.
2. Upon successful authentication and registration, the member device retrieves group SAs and keys from the server with a GDOI `groupkey-pull` exchange.
3. The server adds the member to the membership for the group.
4. Group members exchange packets encrypted with group SA keys.

The server periodically sends SA and key refreshes to group members with rekey (GDOI `groupkey-push`) messages. Rekey messages are sent before SAs expire; this ensures that valid keys are available for encrypting traffic between group members.

The server also sends rekey messages to provide new keys to members when there is a change in group membership or when the group SA has changed.

Understanding Group VPNv1 Server-Member Communication

Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices. Server-member communication allows the server to send GDOI `groupkey-push` messages to members. If server-member communication is not configured for the group, members can send GDOI `groupkey-pull` messages to register and reregister with the server, but the server is not able to send rekey messages to members.

Server-member communication is configured for the group by using the `server-member-communication` *configuration statement* at the [`edit security group-vpn server`] hierarchy. The following options can be defined:

- Encryption algorithm used for communications between the server and member. You can specify 3des-cbc, aes-128-cbc, aes-192-cbc, aes-256-cbc, or des-cbc. There is no default algorithm.
- Authentication algorithm (md5 or sha1) used to authenticate the member to the server. There is no default algorithm.
- Whether the server sends unicast or multicast rekey messages to group members and parameters related to the communication type.
- Interval at which the server sends heartbeat messages to the group member. This allows the member to determine whether the server has rebooted, which would require the member to reregister with the server. The default is 300 seconds.
- Lifetime for the key encryption key (KEK). The default is 3600 seconds.

Configuring server-member communication is necessary for the group server to send rekey messages to members, but there might be situations in which this behavior is not desired. For example, if group members are dynamic peers (such as in a home office), the devices are not always up and the IP address of a device might be different each time it is powered up. Configuring server-member communication for a group of dynamic peers can result in unnecessary transmissions by the server. If you want IKE Phase 1 SA negotiation to always be performed to protect GDOI negotiation, do not configure server-member communication.

If server-member communication for a group is not configured, the membership list displayed by the `show security group-vpn server registered-members` command shows group members who have registered with the server; members can be active or not. When server-member communication for a group is configured, the group membership list is cleared. If the communication type is configured as unicast, the `show security group-vpn server registered-members` command shows only active members. If the communication type is configured as multicast, the `show security group-vpn server registered-members` command shows members who have registered with the server after the configuration; the membership list does not necessarily represent active members because members might drop out after registration.

Understanding Group VPNv1 Group Key Operations

This topic contains the following sections:

Group Keys

The group server maintains a database to track the relationship among VPN groups, group members, and group keys. There are two kinds of group keys that the server downloads to members:

- Key Encryption Key (KEK)—Used to encrypt rekey messages. One KEK is supported per group.
- Traffic Encryption Key (TEK)—Used to encrypt and decrypt IPsec data traffic between group members.

The key associated with an SA is accepted by a group member only if there is a matching scope policy configured on the member. An accepted key is installed for the group VPN, whereas a rejected key is discarded.

Rekey Messages

If the group is configured for server-member communications, the server periodically sends SA and key refreshes to group members with rekey (GDOI `groupkey-push`) messages. Rekey messages are sent before SAs expire; this ensures that valid keys are available for encrypting traffic between group members.

The server also sends rekey messages to provide new keys to members when there is a change in group membership or the group SA has changed (for example, a group policy is added or deleted).

Server-member communications options must be configured on the server to allow the server to send rekey messages to group members. These options specify the type of message and the intervals at which the messages are sent, as explained in the following sections:

There are two types of rekey messages:

- Unicast rekey messages—The group server sends one copy of the rekey message to each group member. Upon receipt of the rekey message, members must send an acknowledgment (ACK) to the server. If the server does not receive an ACK from a member (including retransmission of rekey messages), the server considers the member to be inactive and removes it from the membership list. The server stops sending rekey messages to the member.

  The `number-of-retransmission` and `retransmission-period` configuration statements for server-member communications control the resending of rekey messages by the server when no ACK is received from a member.

- Multicast rekey messages—The group server sends one copy of the rekey message from the specified outgoing interface to the configured multicast group address. Members do not send acknowledgment of receipt of multicast rekey messages. The registered membership list does not necessarily represent active members because members might drop out after initial registration. All members of the group must be configured to support multicast messages.

  IP multicast protocols must be configured to allow delivery of multicast traffic in the network. For detailed information about configuring multicast protocols on Juniper Networks devices, see Multicast Protocols User Guide .

The interval at which the server sends rekey messages is calculated based on the values of the lifetime-seconds and activation-time-delay configuration statements at the [edit security group-vpn server group] hierarchy. The interval is calculated as lifetime-seconds minus 4*(activation-time-delay).

The lifetime-seconds for the KEK is configured as part of the server-member communications; the default is 3600 seconds. The lifetime-seconds for the TEK is configured for the IPsec proposal; the default is 3600 seconds. The activation-time-delay is configured for the group on the server; the default is 15 seconds. Using the default values for lifetime-seconds and activation-time-delay, the interval at which the server sends rekey messages is 3600 minus 4*15, or 3540 seconds.

Member Registration

If a group member does not receive a new SA key from the server before the current key expires, the member must reregister with the server and obtain updated keys with a GDOI groupkey-pull exchange. In this case, the interval at which the server sends rekey messages is calculated as follows: lifetime-seconds minus 3*(activation-time-delay). Using the default values for lifetime-seconds and activation-time-delay, the interval at which the server sends rekey messages is 3600 minus 3*15, or 3555 seconds.

Member reregistration can occur for the following reasons:

- The member detects a server reboot by the absence of heartbeats received from the server.

- The rekey message from the group server is lost or delayed, and the TEK lifetime has expired.

Key Activation

When a member receives a new key from the server, it waits a period of time before using the key for encryption. This period of time is determined by the activation-time-delay *configuration statement* and whether the key is received through a rekey message sent from the server or as a result of the member reregistering with the server.

If the key is received through a rekey message sent from the server, the member waits 2*(activation-time-delay) seconds before using the key. If the key is received through member reregistration, the member waits the number of seconds specified by the activation-time-delay value.

A member retains the two most recent keys sent from the server for each group SA installed on the member. Both keys can be used for decryption, while the most recent key is used for encryption. The previous key is removed the number of seconds specified by the activation-time-delay value after the new key is activated.

The default for the activation-time-delay configuration statement is 15 seconds. Setting this time period too small can result in a packet being dropped at a remote group member before the new key is installed. Consider the network topology and system transport delays when you change the activation-time-delay value. For unicast transmissions, the system transport delay is proportional to the number of group members.

A group VPNv1 server can send multiple traffic encryption keys (TEKs) to a group VPNv1 member in response to a `groupkey-pull` request. The following describes how the group VPNv1 member handles the existing TEK and the TEKs it receives from the server:

- If the group VPNv1 member receives two or more TEKs, it holds the most recent two TEKs and deletes the existing TEK. Of the two held TEKs, the older TEK is activated immediately, and the newer TEK is activated after the `activation-time-delay` configured on the group VPNv1 server has elapsed (the default is 15 seconds).

- If the group VPNv1 member receives only one TEK, or if it receives a TEK through a `groupkey-push` message from the server, the existing TEK is not deleted until the hard lifetime expires. The lifetime is not shortened for the existing TEK.

The group VPNv1 member still installs a received TEK even if the TEK lifetime is less than two times the `activation-time-delay` value.

Understanding Group VPNv1 Heartbeat Messages

When server-member communication is configured, the group VPNv1 server sends heartbeat messages to members at specified intervals (the default interval is 300 seconds). The heartbeat mechanism allows members to reregister with the server if the specified number of heartbeats is not received. For example, members will not receive heartbeat messages during a server reboot. When the server has rebooted, members reregister with the server.

Heartbeats are transmitted through `groupkey-push` messages. The sequence number is incremented on each heartbeat message, which protects members from reply attacks. Unlike rekey messages, heartbeat messages are not acknowledged by recipients and are not retransmitted by the server.

Heartbeat messages contain the following information:

- Current state and configuration of the keys on the server

- Relative time, if antireplay is enabled

By comparing the information in the heartbeats, a member can detect whether it has missed server information or rekey messages. The member reregisters to synchronize itself with the server.

Heartbeat messages can increase network congestion and cause unnecessary member reregistrations. Thus, heartbeat detection can be disabled on the member if necessary.

Understanding Group VPNv1 Server-Member Colocation Mode

Group server and group member functions are separate and do not overlap. The server and member functions can coexist in the same physical device, which is referred as colocation mode. In colocation mode, there is no change in terms of functionality and behavior of the server or a member, but the server and member each need to be assigned different IP addresses so that packets can be delivered properly. In colocation mode, there can be only one IP address assigned to the server and one IP address assigned to the member across groups.

SEE ALSO

IPsec Overview | 20

Understanding IKE and IPsec Packet Processing | 170

Group VPNv1 Configuration Overview | 713

Group VPNv1 Configuration Overview

This topic describes the main tasks for configuring group VPNv1.

On the group server, configure the following:

1. IKE Phase 1 negotiation. Use the [edit security group-vpn server ike] hierarchy to configure the IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv2 " on page 765.
2. Phase 2 IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv1" on page 715.
3. VPN group. See "Group VPNv1 Configuration Overview" on page 713.

On the group member, configure the following:

1. IKE Phase 1 negotiation. Use the [edit security group-vpn member ike] hierarchy to configure IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv1 " on page 714.

2. Phase 2 IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv1" on page 715.

3. Scope policy that determines which group policies are installed on the member. See "Understanding Dynamic Policies for Group VPNv1" on page 715.

To prevent packet fragmentation issues, we recommend that the interface used by the group member to connect to the MPLS network be configured for a maximum transmission unit (MTU) size no larger than 1400 bytes. Use the set *interface* mtu configuration statement to set the MTU size.

The VPN group is configured on the server with the group *configuration statement* at the [edit security group-vpn server] hierarchy.

The group information consists of the following information:

- Group identifier—A value between 1 and 65,535 that identifies the VPN group. The same group identifier must be configured on the group member for Autokey IKE.

- Group members, as configured with the `ike-gateway` configuration statement. There can be multiple instances of this configuration statement, one for each member of the group.

- IP address of the server (the loopback interface address is recommended).

- Group policies—Policies that are to be downloaded to members. Group policies describe the traffic to which the SA and keys apply. See "Understanding Dynamic Policies for Group VPNv1" on page 715.

- Server-member communication—Optional configuration that allows the server to send rekey messages to members. See "Group VPNv1 Overview" on page 703.

- Antireplay—Optional configuration that detects packet interception and replay. See "Understanding Antireplay for Group VPNv1" on page 716.

Understanding IKE Phase 1 Configuration for Group VPNv1

An IKE Phase 1 SA between the group server and a group member establishes a secure channel in which to negotiate IPsec SAs that are shared by a group. For standard IPsec VPNs on Juniper Networks security devices, Phase 1 SA configuration consists of specifying an IKE proposal, policy, and gateway. For group VPNv1, the IKE Phase 1 SA configuration is similar to the configuration for standard IPsec VPNs, but is performed at the `[edit security group-vpn]` hierarchy.

In the IKE proposal configuration, you set the authentication method and the authentication and encryption algorithms that will be used to open a secure channel between participants. In the IKE policy configuration, you set the mode (main or aggressive) in which the Phase 1 channel will be negotiated, specify the type of key exchange to be used, and reference the Phase 1 proposal. In the IKE gateway configuration, you reference the Phase 1 policy.

Because Group VPNv2 only supports strong algorithms, the `sha-256` authentication algorithm option is supported for Group VPNv1 members on SRX100, SRX110, SRX210, SRX220, SRX240, SRX550, and SRX650 devices. When Group VPNv1 members interoperate with Group VPNv2 servers, this option must be configured on the Group VPNv1 members with the `edit security group-vpn member ike proposal proposal-name authentication-algorithm sha-256` command. On the Group VPNv2 server, `authentication-algorithm sha-256` must be configured for IKE proposals and `authentication-algorithm hmac-sha-256-128` must be configured for IPsec proposals.

If an IKE gateway on a Group VPNv1 member is configured with more than one gateway address, the error message “Only one remote address is allowed to be configured per IKE gateway configuration” is displayed when the configuration is committed.

The IKE Phase 1 configuration on the group server must match the IKE Phase 1 configuration on group members.

Understanding IPsec SA Configuration for Group VPNv1

After the server and member have established a secure and authenticated channel in Phase 1 negotiation, they proceed through Phase 2. Phase 2 negotiation establishes the IPsec SAs that are shared by group members to secure data that is transmitted among members. While the IPsec SA configuration for group VPN is similar to the configuration for standard VPNs, a group member does not need to negotiate the SA with other group members.

Phase 2 IPsec configuration for group VPNv1 consists of the following information:

- A proposal for the security protocol, authentication, and encryption algorithm to be used for the SA. The IPsec SA proposal is configured on the group server with the `proposal` *configuration statement* at the [`edit security group-vpn server ipsec`] hierarchy.

- A group policy that references the proposal. A group policy specifies the traffic (protocol, source address, source port, destination address, and destination port) to which the SA and keys apply. The group policy is configured on the server with the `ipsec-sa` configuration statement at the [`edit security group-vpn server group`] hierarchy.

- An Autokey IKE that references the group identifier, the group server (configured with the `ike-gateway` configuration statement), and the interface used by the member to connect to the group. The Autokey IKE is configured on the member with the `ipsec vpn` configuration statement at the [`edit security group-vpn member`] hierarchy.

Understanding Dynamic Policies for Group VPNv1

The group server distributes group SAs and keys to members of a specified group. All members that belong to the same group can share the same set of IPsec SAs. But not all SAs configured for a group are installed on every group member. The SA installed on a specific member is determined by the policy associated with the group SA and the security policies configured on the member.

In a VPN group, each group SA and key that the server pushes to a member is associated with a group policy. The group policy describes the traffic on which the key should be used, including protocol, source address, source port, destination address, and destination port.

Group policies that are identical (configured with the same source address, destination address, source port, destination port, and protocol values) cannot exist for a single group. An error is returned if you attempt to commit a configuration that contains identical group policies for a group. If this is the case, you must delete one of the identical group policies.

On a group member, a scope policy must be configured that defines the scope of the group policy downloaded from the server. A group policy distributed from the server is compared against the scope policies configured on the member. For a group policy to be installed on the member, the following conditions must be met:

- Any addresses specified in the group policy must be within the range of addresses specified in the scope policy.

- The source port, destination port, and protocol specified in the group policy must match those configured in the scope policy.

A group policy that is installed on a member is called a dynamic policy.

A scope policy can be part of an ordered list of security policies for a specific from-zone and to-zone context. Junos OS performs a security policy lookup on incoming packets starting from the top of the ordered list.

Depending on the position of the scope policy within the ordered list of security policies, there are several possibilities for dynamic policy lookup:

- If the incoming packet matches a security policy before the scope policy is considered, dynamic policy lookup does not occur.

- If an incoming policy matches a scope policy, the search process continues for a matching dynamic policy. If there is a matching dynamic policy, that policy action (permit) is performed. If there is no matching dynamic policy, the search process continues to search the policies below the scope policy.

  In this release, only the `tunnel` action is allowed for a scope policy. Other actions are not supported.

You configure a scope policy on a group member by using the `policies` *configuration statement* at the [`edit security`] hierarchy. Use the `ipsec-group-vpn` configuration statement in the permit tunnel rule to reference the group VPN; this allows group members to share a single SA.

SEE ALSO

Security Policies Overview

Understanding Security Policy Ordering

Example: Configuring a Security Policy to Permit or Deny All Traffic

Understanding Antireplay for Group VPNv1

Antireplay is an IPsec feature that can detect when a packet is intercepted and then replayed by attackers. Antireplay is enabled by default for group VPNs but can be disabled for a group with the `no-anti-replay` *configuration statement*.

When antireplay is enabled, the group server synchronizes the time between the group members. Each IPsec packet contains a timestamp. The group member checks whether the packet's timestamp falls within the configured `anti-replay-time-window` value (the default is 100 seconds). A packet is dropped if the timestamp exceeds the value.

SEE ALSO

IPsec Overview | 20

Understanding IKE and IPsec Packet Processing | 170

Example: Configuring Group VPNv1 Server and Members

IN THIS SECTION

This example shows how to configure group VPNv1 to extend IPsec architecture to support SAs that are shared by a group of security devices. Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices.

Requirements

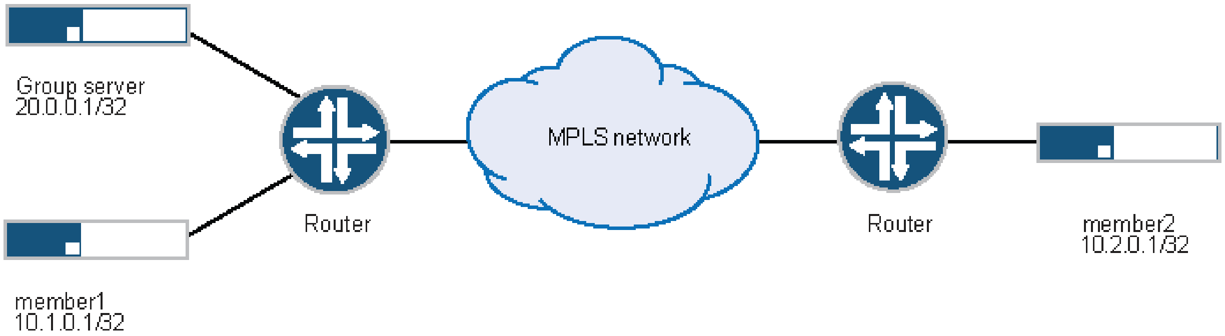
Before you begin:

- Configure the Juniper Networks security devices for network communication.
- Configure network interfaces on server and member devices. See Interfaces User Guide for Security Devices.

Overview

In Figure 50 on page 718, a group VPN consists of two member devices (member1 and member2) and a group server (the IP address of the loopback interface on the server is 20.0.0.1). The group identifier is 1.

Figure 50: Server-Member Configuration Example



The Phase 2 group VPN SAs must be protected by a Phase 1 SA. Therefore, the group VPN configuration must include configuring IKE Phase 1 negotiations on both the group server and the group members. In addition, the same group identifier must be configured on both the group server and the group members.

Group policies are configured on the group server. All group policies configured for a group are downloaded to group members. Scope policies configured on a group member determine which group policies are actually installed on the member. In this example, the following group policies are configured on the group server for downloading to all group members:

- p1—Allows all traffic from 10.1.0.0/16 to 10.2.0.0./16
- p2—Allows all traffic from 10.2.0.0./16 to 10.1.0.0/16
- p3—Allows multicast traffic from 10.1.1.1/32

The member1 device is configured with scope policies that allow all unicast traffic to and from the 10.0.0.0/8 subnetwork. There is no scope policy configured on member1 to allow multicast traffic; therefore, the SA policy p3 is not installed on member1.

The member2 device is configured with scope policies that drop traffic from 10.1.0.0/16 from the trust zone to the untrust zone and to 10.1.0.0/16 from the untrust zone to the trust zone. Therefore the SA policy p2 is not installed on member2.

Configuration

IN THIS SECTION

Configuring the Group Server

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces lo0 unit 0 family inet address 20.0.0.1/32
set security group-vpn server ike proposal srv-prop authentication-method pre-shared-keys
set security group-vpn server ike proposal srv-prop dh-group group2
set security group-vpn server ike proposal srv-prop authentication-algorithm sha1
set security group-vpn server ike proposal srv-prop encryption-algorithm 3des-cbc
set security group-vpn server ike policy srv-pol mode main
set security group-vpn server ike policy srv-pol proposals srv-prop
set security group-vpn server ike policy srv-pol pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike gateway gw1 ike-policy srv-pol
set security group-vpn server ike gateway gw1 address 10.1.0.1
set security group-vpn server ike gateway gw2 ike-policy srv-pol
set security group-vpn server ike gateway gw2 address 10.2.0.1
set security group-vpn server ipsec proposal group-prop authentication-algorithm hmac-sha1-96
set security group-vpn server ipsec proposal group-prop encryption-algorithm 3des-cbc
set security group-vpn server ipsec proposal group-prop lifetime-seconds 3600
set security group-vpn server group grp1 group-id 1
set security group-vpn server group grp1 ike-gateway gw1
set security group-vpn server group grp1 ike-gateway gw2
set security group-vpn server group grp1 anti-replay-time-window 120
set security group-vpn server group grp1 server-address 20.0.0.1
set security group-vpn server group grp1 ipsec-sa group-sa proposal group-prop
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 source 10.1.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 destination
10.2.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 source-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 protocol 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 source 10.2.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 destination
10.1.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 source-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 protocol 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 source 10.1.1.1/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 destination
239.1.1.1/32
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 source-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 protocol 0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the group server:

1. Configure the loopback address on the device.

   ```
   [edit]
   user@host# edit interfaces
   user@host# set lo0 unit 0 family inet address 20.0.0.1/32
   ```

2. Configure IKE Phase 1 SA (this configuration must match the Phase 1 SA configured on the group members).

   ```
   [edit security group-vpn server ike proposal srv-prop]
   user@host# set authentication-method pre-shared-keys
   user@host# set dh-group group2
   user@host# set authentication-algorithm sha1
   user@host# set encryption-algorithm 3des-cbc
   ```

3. Define the IKE policy and set the remote gateways.

   ```
   [edit security group-vpn server ike]
   user@host# set policy srv-pol mode main proposals srv-prop pre-shared-key ascii-text "$ABC123"
   user@host# set gateway gw1 ike-policy srv-pol address 10.1.0.1
   user@host# set gateway gw2 ike-policy srv-pol address 10.2.0.1
   ```

4. Configure the Phase 2 SA exchange.

   ```
   [edit security group-vpn server ipsec proposal group-prop]
   user@host# set authentication-algorithm hmac-sha1-96
   ```

```
user@host# set encryption-algorithm 3des-cbc
user@host# set lifetime-seconds 3600
```

5. Configure the group identifier and IKE gateway.

```
[edit security group-vpn server group grp1]
user@host# set group-id 1
user@host# set ike-gateway gw1
user@host# set ike-gateway gw2
user@host# set anti-replay-time-window 120 server-address 20.0.0.1
```

6. Configure server-to-member communications.

```
[edit security group-vpn server group grp1]
user@host# set server-member-communication communication-type unicast encryption-algorithm
aes-128-cbc sig-hash-algorithm md5 certificate “srv-cert”
```

7. Configure the group policies to be downloaded to group members.

```
[edit security group-vpn server group grp1 ipsec-sa group-sa]
user@host# set proposal group-prop match-policy p1 source 10.1.0.0/16 destination 10.2.0.0/16
source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p2 source 10.2.0.0/16 destination 10.1.0.0/16
source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p3 source 10.1.1.1/16 destination
239.1.1.1/32 source-port 0 destination-port 0 protocol 0
```

Results

From configuration mode, confirm your configuration by entering the `show security group-vpn server` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security group-vpn server
ike {
    proposal srv-prop {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm sha1;
        encryption-algorithm 3des-cbc;
    }
    policy srv-pol {
        mode main;
        proposals srv-prop;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
    }
    gateway gw1 {
        ike-policy srv-pol;
        address 10.1.0.1;
    }
    gateway gw2 {
        ike-policy srv-pol;
        address 10.2.0.1;
    }
}
    ipsec {
        proposal group-prop {
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 3600;
        }
    }
    group grp1 {
        group-id 1;
        ike-gateway gw1;
        ike-gateway gw2;
        anti-replay-time-window 120;
        server-address 20.0.0.1;
        ipsec-sa group-sa {
            proposal group-prop;
            match-policy p1 {
                source 10.1.0.0/16;
                destination 10.2.0.0/16;
                source-port 0;
                destination-port 0;
                protocol 0;
            }
            match-policy p2 {
                source 10.2.0.0/16;
                destination 10.1.0.0/16;
                source-port 0;
                destination-port 0;
                protocol 0;
            }
            match-policy p3 {
                source 10.1.1.1/16;
                destination 239.1.1.1/32;
                source-port 0;
                destination-port 0;
                protocol 0;
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Member1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security group-vpn member ike proposal prop1 authentication-method pre-shared-keys
set security group-vpn member ike proposal prop1 dh-group group2
set security group-vpn member ike proposal prop1 authentication-algorithm sha1
set security group-vpn member ike proposal prop1 encryption-algorithm 3des-cbc
set security group-vpn member ike policy pol1 mode main
set security group-vpn member ike policy pol1 proposals prop1
set security group-vpn member ike policy pol1 pre-shared-key ascii-text "$ABC123"
set security group-vpn member ike gateway g1 ike-policy pol1
set security group-vpn member ike gateway g1 address 20.0.0.1
set security group-vpn member ike gateway g1 local-address 10.1.0.1
set security group-vpn member ipsec vpn v1 ike-gateway g1
set security group-vpn member ipsec vpn v1 group-vpn-external-interface ge-0/1/0
set security group-vpn member ipsec vpn v1 group 1
set security address-book book1 address 10_subnet 10.0.0.0/8
set security address-book book1 attach zone trust
set security address-book book2 address 10_subnet 10.0.0.0/8
set security address-book book2 attach zone untrust
set security policies from-zone trust to-zone untrust policy scope1 match source-address
```

```
10_subnet
set security policies from-zone trust to-zone untrust policy scope1 match destination-address
10_subnet
set security policies from-zone trust to-zone untrust policy scope1 match application any
set security policies from-zone trust to-zone untrust policy scope1 then permit tunnel ipsec-
group-vpn v1
set security policies from-zone untrust to-zone trust policy scope1 match source-address
10_subnet
set security policies from-zone untrust to-zone trust policy scope1 match destination-address
10_subnet
set security policies from-zone untrust to-zone trust policy scope1 match application any
set security policies from-zone untrust to-zone trust policy scope1 then permit tunnel ipsec-
group-vpn v1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure member1:

1. Configure Phase 1 SA (this configuration must match the Phase 1 SA configured on the group server).

```
[edit security group-vpn member ike proposal prop1]
user@member1# set authentication-method pre-shared-keys
user@member1# set dh-group group2
user@member1# set authentication-algorithm sha1
user@member1# set encryption-algorithm 3des-cbc
```

2. Define the IKE policy and set the remote gateways.

```
[edit security group-vpn member ike]
user@member1# set policy pol1 mode main proposals prop1 pre-shared-key ascii-text "$ABC123"
user@member1# set gateway g1 ike-policy pol1 address 20.0.0.1 local-address 10.1.0.1
```

3. Configure the group identifier, IKE gateway, and interface for member1.

```
[edit security group-vpn member ipsec]
user@member1# set vpn v1 group 1 ike-gateway g1 group-vpn-external-interface ge-0/1/0
```

To prevent packet fragmentation issues, we recommend that the interface used by the group members to connect to the MPLS network be configured for an MTU size no larger than 1400 bytes. Use the `set interface mtu` configuration statement to set the MTU size.

4. Create address books and attach zones to them.

```
[edit security address-book book1]
user@member1# set address 10_subnet 10.0.0.0/8
user@member1# set attach zone trust
```

```
[edit security address-book book2]
user@member1# set address 10_subnet 10.0.0.0/8
user@member1# set attach zone untrust
```

5. Configure a scope policy from the trust zone to the untrust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone trust to-zone untrust]
user@member1# set policy scope1 match source-address 10_subnet destination-address 10_subnet
application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

6. Configure a scope policy from the untrust zone to the trust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone untrust to-zone trust]
user@member1# set policy scope1 match source-address 10_subnet destination-address 10_subnet
application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

Results

From configuration mode, confirm your configuration by entering the `show security group-vpn member` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@member1# show security group-vpn member
ike {
    proposal prop1 {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm sha1;
        encryption-algorithm 3des-cbc;
    }
    policy pol1 {
        mode main;
        proposals prop1;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
    }
    gateway g1 {
        ike-policy pol1;
        address 20.0.0.1;
        local-address 10.1.0.1;
    }
}
    ipsec {
        vpn v1 {
            ike-gateway g1;
            group-vpn-external-interface ge-0/1/0;
            group 1;
        }
    }
```

```
[edit]
user@member1# show security policies
from-zone trust to-zone trust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
    from-zone trust to-zone untrust {
        policy scope1 {
            match {
                source-address 10_subnet;
                destination-address 10_subnet;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v1;
                    }
                }
            }
        }
        policy default-permit {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone untrust to-zone trust {
        policy scope1 {
            match {
                source-address 10_subnet;
                destination-address 10_subnet;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v1;
                    }
                }
            }
        }
        policy default-deny {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                deny;
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Member2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security group-vpn member ike proposal prop2 authentication-method pre-shared-keys
set security group-vpn member ike proposal prop2 authentication-method pre-shared-keys
set security group-vpn member ike proposal prop2 dh-group group2
set security group-vpn member ike proposal prop2 authentication-algorithm sha1
set security group-vpn member ike proposal prop2 encryption-algorithm 3des-cbc
set security group-vpn member ike policy pol2 mode main
set security group-vpn member ike policy pol2 proposals prop2
set security group-vpn member ike policy pol2 pre-shared-key ascii-text "$ABC123"
set security group-vpn member ike gateway g2 ike-policy pol2
set security group-vpn member ike gateway g2 address 20.0.0.1
set security group-vpn member ike gateway g2 local-address 10.2.0.1
set security group-vpn member ipsec vpn v2 ike-gateway g2
set security group-vpn member ipsec vpn v2 group-vpn-external-interface ge-0/1/0
set security group-vpn member ipsec vpn v2 group 1
set security address-book book1 address 10_subnet 10.0.0.0/8
set security address-book book1 address 10_1_0_0_16 10.1.0.0/16
set security address-book book1 address multicast_net 239.0.0.0/8
set security address-book book1 attach zone trust
set security address-book book2 address 10_subnet 10.0.0.0/8
set security address-book book2 address 10_1_0_0_16 10.1.0.0/16
set security address-book book2 address multicast_net 239.0.0.0/8
set security address-book book2 attach zone untrust
set security policies from-zone trust to-zone untrust policy deny2 match source-address
10_1_0_0_16
set security policies from-zone trust to-zone untrust policy deny2 match destination-address any
set security policies from-zone trust to-zone untrust policy deny2 match application any
set security policies from-zone trust to-zone untrust policy deny2 then reject
set security policies from-zone trust to-zone untrust policy scope2 match source -address
10_subnet
set security policies from-zone trust to-zone untrust policy scope2 match destination-address
10_subnet
set security policies from-zone trust to-zone untrust policy scope2 match application any
set security policies from-zone trust to-zone untrust policy scope2 then permit tunnel ipsec-
group-vpn v2
set security policies from-zone trust to-zone untrust policy multicast-scope2 match source-
address 10_subnet
set security policies from-zone trust to-zone untrust policy multicast-scope2 match destination-
address multicast-net
set security policies from-zone trust to-zone untrust policy multicast-scope2 match application
any
set security policies from-zone trust to-zone untrust policy multicast-scope2 then permit tunnel
ipsec-group-vpn v2
set security policies from-zone untrust to-zone trust policy deny2 match source-address any set
security policies from-zone untrust to-zone trust policy multicast-scope2 ma tch application any
set security policies from-zone untr
set security policies from-zone untrust to-zone trust policy deny2 match destination-address
10_1_0_0_16
set security policies from-zone untrust to-zone trust policy deny2 match application any
set security policies from-zone untrust to-zone trust policy deny2 then reject
set security policies from-zone untrust to-zone trust policy scope2 match source-address
10_subnet
set security policies from-zone untrust to-zone trust policy scope2 match destination-address
10_subnet
set security policies from-zone untrust to-zone trust policy scope2 match application any
set security policies from-zone untrust to-zone trust policy scope2 then permit tunnel ipsec-
group-vpn v2
set security policies from-zone untrust to-zone trust policy multicast-scope2 match source-
address 10_subnet
```

```
set security policies from-zone untrust to-zone trust policy multicast-scope2 match destination-
address multicast-net
set security policies from-zone untrust to-zone trust policy multicast-scope2 match application
any
set security policies from-zone untrust to-zone trust policy multicast-scope2 then permit tunnel
ipsec-group-vpn v2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure member2:

1. Configure Phase 1 SA (this configuration must match the Phase 1 SA configured on the group server).

   ```
   [edit security group-vpn member ike proposal prop2]
   user@member2# set authentication-method pre-shared-keys
   user@member2# set dh-group group2
   user@member2# set authentication-algorithm sha1
   user@member2# set encryption-algorithm 3des-cbc
   ```

2. Define the IKE policy and set the remote gateway.

   ```
   [edit security group-vpn member ike]
   user@member2# set policy pol2 mode main proposals prop2 pre-shared-key ascii-text "$ABC123"
   user@member2# set gateway g2 ike-policy pol2 address 20.0.0.1 local-address 10.2.0.1
   ```

3. Configure the group identifier, IKE gateway, and interface for member2.

   ```
   [edit security group-vpn member ipsec]
   user@member2# set vpn v2 group 1 ike-gateway g2 group-vpn-external-interface ge-0/1/0
   ```

   To prevent packet fragmentation issues, we recommend that the interface used by the group members to connect to the MPLS network be configured for an MTU size no larger than 1400 bytes. Use the `set` *`interface`* `mtu` configuration statement to set the MTU size.

4. Create an address book and attach it to the trust zone.

```
[edit security address-book book1]
user@member2# set address 10_subnet 10.0.0.0/8
user@member2# set address 10_1_0_0_16 10.1.0.0/16
user@member2# set address multicast_net 239.0.0.0/8
user@member2# set attach zone trust
```

5. Create another address book and attach it to the untrust zone.

```
[edit security address-book book2]
user@member2# set address 10_subnet 10.0.0.0/8
user@member2# set address 10_1_0_0_16 10.1.0.0/16
user@member2# set address multicast_net 239.0.0.0/8
user@member2# set attach zone untrust
```

6. Configure a scope policy from the trust zone to the untrust zone that blocks traffic from 10.1.0.0/16.

```
[edit security policies from-zone trust to-zone untrust]
user@member2# set policy deny2 match source-address 10_1_0_0_16 destination-address any
application any
user@member2# set policy deny2 then reject
user@member2# set policy scope2 match source-address 10_subnet destination-address 10_subnet
application any
user@member2# set policy scope2 then permit tunnel ipsec-group-vpn v2
user@member2# set policy multicast-scope2 match source-address 10_subnet destination-address
multicast-net application any
user@member2# set policy multicast-scope2 then permit tunnel ipsec-group-vpn v2
```

7. Configure a scope policy from the untrust zone to the trust zone that blocks traffic to 10.1.0.0/16.

```
[edit security policies from-zone untrust to-zone trust]
user@member2# set policy deny2 match source-address any destination-address 10_1_0_0_16
application any
user@member2# set policy deny2 then reject
user@member2# set policy scope2 match source-address 10_subnet destination-address 10_subnet
application any
user@member2# set policy scope2 then permit tunnel ipsec-group-vpn v2
user@member2# set policy multicast-scope2 match source-address 10_subnet destination-address
```

```
multicast-net application any
user@member2# set policy multicast-scope2 then permit tunnel ipsec-group-vpn v2
```

Results

From configuration mode, confirm your configuration by entering the `show security group-vpn member` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@member2# show security group-vpn member
ike {
    proposal prop2 {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm sha1;
        encryption-algorithm 3des-cbc;
    }
    policy pol2 {
        mode main;
        proposals prop2;
        pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
    }
    gateway g2 {
        ike-policy pol2;
        address 20.0.0.1;
        local-address 10.2.0.1;
    }
}
    ipsec {
        vpn v2 {
            ike-gateway g2;
            group-vpn-external-interface ge-0/1/0;
            group 1;
        }
    }
```

```
[edit]
user@member2# show security policies
from-zone trust to-zone trust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
    from-zone trust to-zone untrust {
        policy deny2 {
            match {
                source-address 10_1_0_0_16;
                destination-address any;
                application any;
            }
            then {
                reject;
            }
        }
        policy scope2 {
            match {
                source-address 10_subnet;
                destination-address 10_subnet;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v2;
                    }
                }
            }
        }
        policy multicast-scope2 {
            match {
                source-address 10_subnet;
                destination-address multicast-net;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v2;
                    }
                }
            }
        }
        policy default-permit {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
    from-zone untrust to-zone trust {
        policy deny2 {
            match {
                source-address any;
                destination-address 10_1_0_0_16;
                application any;
            }
            then {
                reject;
            }
        }
        policy scope2 {
            match {
                source-address 10_subnet;
                destination-address 10_subnet;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v2;
                    }
                }
            }
        }
        policy multicast-scope2 {
            match {
                source-address 10_subnet;
                destination-address multicast-net;
                application any;
            }
            then {
                permit {
                    tunnel {
                        ipsec-group-vpn v2;
                    }
                }
            }
        }
        policy default-deny {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                deny;
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform this task:

Verifying Dynamic Policies for Member1

Purpose

View the dynamic policies installed on member1.

Action

After the group server downloads keys to member1, enter the `show security dynamic-policies` command from operational mode.

```
user@member1> show security dynamic-policies
Policy: scope1-0001, action-type: permit, State: enabled, Index: 1048580,AI: disabled, Scope
Policy: 4
  Policy Type: Dynamic
  Sequence number: 1
  From zone: untrust, To zone: trust
  Source addresses: 10.1.0.0/16
  Destination addresses: 10.2.0.0/16
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope1-0001, action-type: permit, State: enabled, Index: 1048581,AI: disabled, Scope
Policy: 5
  Policy Type: Dynamic
  Sequence number: 2
  From zone: trust, To zone: untrust
  Source addresses: 10.1.0.0/16
  Destination addresses: 10.2.0.0/16
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
```

Meaning

The multicast policy p3 from the server is not installed on member1 because there is no scope policy configured on member1 that allows multicast traffic.

Verifying Dynamic Policies for Member2

Purpose

View the dynamic policies installed on member 2.

Action

After the group server downloads keys to member2, enter the `show security dynamic-policies` command from operational mode.

```
user@member2> show security dynamic-policies
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048580,AI: disabled, Scope
Policy: 4
  Policy Type: Dynamic
  Sequence number: 1
  From zone: untrust, To zone: trust
  Source addresses: 10.1.0.0/16
  Destination addresses: 10.2.0.0/16
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048580,AI: disabled, Scope
Policy: 4
  Policy Type: Dynamic
  Sequence number: 1
  From zone: untrust, To zone: trust
  Source addresses: 10.1.1.1/32
  Destination addresses: 239.1.1.1/32
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048581,AI: disabled, Scope
Policy: 5
  Policy Type: Dynamic
  Sequence number: 2
  From zone: trust, To zone: untrust
  Source addresses: 10.2.0.0/16/0
  Destination addresses: 10.1.0.0/16
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048581,AI: disabled, Scope
Policy: 5
  Policy Type: Dynamic
  Sequence number: 2
  From zone: trust, To zone: untrust
  Source addresses: 10.1.1.1/32
  Destination addresses: 239.1.1.1/32
  Application: Unknown
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
```

Meaning

The policy p2 (for traffic from 10.1.0.0/16 to 10.2.0.0/16) from the server is not installed on member2, because it matches the deny2 security policy configured on member2.

Example: Configuring Group VPNv1 Server-Member Communication for Unicast Rekey Messages

IN THIS SECTION

This example shows how to enable the server to send unicast rekey messages to group members to ensure that valid keys are available for encrypting traffic between group members. Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices.

Requirements

Before you begin:

- Configure the group server and members for IKE Phase 1 negotiation.
- Configure the group server and members for Phase 2 IPsec SA.
- Configure the group `g1` on the group server.

Overview

In this example, you specify the following server-member communication parameters for group `g1`:

- The server sends unicast rekey messages to group members.
- 3des-cbc is used to encrypt traffic between the server and members.
- sha1 is used for member authentication.

Default values are used for server heartbeats, KEK lifetime, and retransmissions.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure server-member communication:

1. Set the communications type.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set communications-type unicast
```

2. Set the encryption algorithm.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set encryption-algorithm 3des-cbc
```

3. Set the member authentication.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set sig-hash-algorithm sha1
```

Verification

To verify the configuration is working properly, enter the `show security group-vpn server group g1 server-member-communication` command.

SEE ALSO

Understanding IKE and IPsec Packet Processing | 170

Example: Configuring Group VPNv1 Server-Member Communication for Multicast Rekey Messages

IN THIS SECTION

This example shows how to enable the server to send multicast rekey messages to group members to ensure that valid keys are available for encrypting traffic between group members. Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices.

Requirements

Before you begin:

- Configure the group server and members for IKE Phase 1 negotiation and Phase 2 IPsec SA. See "Example: Configuring Group VPNv1 Server and Members" on page 717 or "Example: Configuring Group VPNv1 with Server-Member Colocation" on page 744.
- Configure ge-0/0/1.0, which is the interface the server will use for sending multicast messages. See Junos OS Routing Protocols Library.
- Configure the multicast group address 226.1.1.1. See Junos OS Routing Protocols Library.

  IP multicast protocols must be configured to allow delivery of multicast traffic in the network. This example does not show multicast configuration.

Overview

In this example, you specify the following server-member communication for group `g1`:

- The server sends multicast rekey messages to group members by means of multicast address 226.1.1.1 and interface ge-0/0/1.0.
- 3des-cbc is used to encrypt traffic between the server and members.
- sha1 is used for member authentication.

Default values are used for server heartbeats, KEK lifetime, and retransmissions.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security group-vpn server group g1 server-member-communication communication-type multicast
set security group-vpn server group g1 server-member-communication multicast-group 226.1.1.1
set security group-vpn server group g1 server-member-communication multicast-outgoing-interface
ge-0/0/1.0
set security group-vpn server group g1 server-member-communication encryption-algorithm 3des-cbc
set security group-vpn server group g1 server-member-communication sig-hash-algorithm sha1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure configure server-member communication for multicast rekey messages:

1. Set the communications type.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set communication-type multicast
```

2. Set the multicast group.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set multicast-group 226.1.1.1
```

3. Set the interface for outgoing multicast messages.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set multicast-outgoing-interface ge-0/0/1.0
```

4. Set the encryption algorithm.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set encryption-algorithm 3des-cbc
```

5. Set the member authentication.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set sig-hash-algorithm sha1
```

Results

From configuration mode, confirm your configuration by entering the `show security group-vpn server group g1 server-member-communication` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security group-vpn server group g1 server-member-communication
communication-type multicast;
multicast-group 226.1.1.1;
multicast-outgoing-interface ge-0/0/1.0;
encryption-algorithm 3des-cbc;
sig-hash-algorithm sha1;
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying Server-Member Communication for Multicast Rekey Messages

Purpose

Verify that server-member communication parameters for multicast rekey message are configured properly to ensure that valid keys are available for encrypting traffic between group members.

Action

From operational mode, enter the `show security group-vpn server group g1 server-member-communication` command.

SEE ALSO

Example: Configuring a Group IKE ID for Multiple Users

Understanding IKE and IPsec Packet Processing | 170

Example: Configuring Group VPNv1 with Server-Member Colocation

IN THIS SECTION

This example shows how to configure a device for colocation mode, which allows server and member functions to coexist on the same physical device. Group VPNv1 is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices.

Requirements

Before you begin:

- Configure the Juniper Networks security devices for network communication.

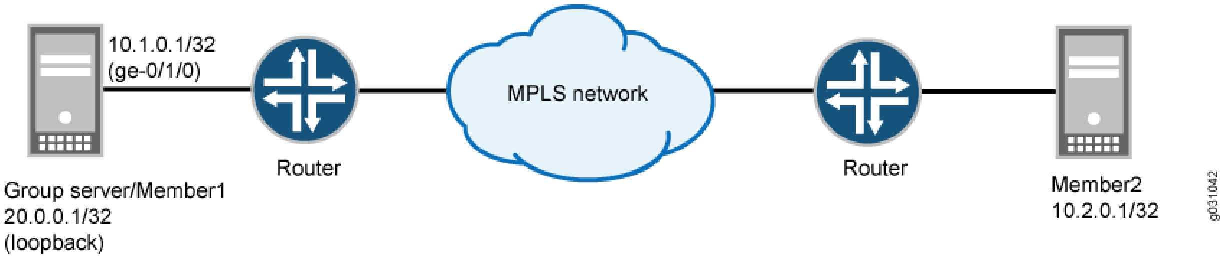
- Configure network interfaces on server and member devices. See Interfaces User Guide for Security Devices.

Overview

When colocation mode is configured, group server and group member functions can coexist in the same device. In colocation mode, the server and member must have different IP addresses so that packets are delivered properly.

In Figure 51 on page 745, a group VPN (group identifier is 1) consists of two members (member1 and member2) and a group server (the IP address of the loopback interface is 20.0.0.1). Note that member1 coexists in the same device as the group server. In this example, the interface that member1 uses to connect to the MPLS network (ge-0/1/0) is assigned the IP address 10.1.0.1/32.

Figure 51: Server-Member Colocation Example



The configuration instructions in this topic describe how to configure the group server-member1 device for colocation mode. To configure member2, see "Example: Configuring Group VPNv1 Server and Members" on page 717.

To prevent packet fragmentation issues, we recommend that the interface used by the group member to connect to the MPLS network be configured for an MTU size no larger than 1400 bytes. Use the `set` *`interface`* `mtu` configuration statement to set the MTU size.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces lo0 unit 0 family inet address 20.0.0.1/32
set interfaces ge-0/1/0 unit 0 family inet address 10.1.0.1/32
set security group-vpn member ike proposal prop1 authentication-method pre-shared-keys
set security group-vpn member ike proposal prop1 dh-group group2
set security group-vpn member ike proposal prop1 authentication-algorithm sha1
set security group-vpn member ike proposal prop1 encryption-algorithm 3des-cbc
set security group-vpn member ike policy pol1 mode main
set security group-vpn member ike policy pol1 proposals prop1
set security group-vpn member ike policy pol1 pre-shared-key ascii-text "$9$c1gr K8-
VYZUHX7UHqmF3Sre"
set security group-vpn member ike gateway g1 ike-policy pol1
set security group-vpn member ike gateway g1 address 20.0.0.1
set security group-vpn member ike gateway g1 local-address 10.1.0.1
set security group-vpn member ipsec vpn v1 ike-gateway g1
set security group-vpn member ipsec vpn v1 group-vpn-external-interface ge-0/1/0
set security group-vpn member ipsec vpn v1 group 1
set security group-vpn server ike proposal srv-prop authentication-method pre-shared-keys
set security group-vpn server ike proposal srv-prop dh-group group2
set security group-vpn server ike proposal srv-prop authentication-algorithm sha1
set security group-vpn server ike proposal srv-prop encryption-algorithm 3des-cbc
set security group-vpn server ike policy srv-pol mode main
set security group-vpn server ike policy srv-pol proposals srv-prop
set security group-vpn server ike policy srv-pol pre-shared-key ascii-text "$9$c 1grK8-
VYZUHX7UHqmF3Sre"
set security group-vpn server ike gateway gw1 ike-policy srv-pol
set security group-vpn server ike gateway gw1 address 10.1.0.1
set security group-vpn server ike gateway gw2 ike-policy srv-pol
set security group-vpn server ike gateway gw2 address 10.2.0.1
set security group-vpn server ipsec proposal group-prop authentication-algorithm hmac-sha1-96
set security group-vpn server ipsec proposal group-prop encryption-algorithm 3des-cbc
set security group-vpn server ipsec proposal group-prop lifetime-seconds 3600
set security group-vpn server group grp1 group-id 1
set security group-vpn server group grp1 ike-gateway gw1
set security group-vpn server group grp1 ike-gateway gw2
set security group-vpn server group grp1 anti-replay-time-window 120
set security group-vpn server group grp1 server-address 20.0.0.1
set security group-vpn server group grp1 server-member-communication communication-type unicast
set security group-vpn server group grp1 server-member-communication encryption-algorithm
aes-128-cbc
set security group-vpn server group grp1 server-member-communication sig-hash-algorithm md5
set security group-vpn server group grp1 server-member-communication certificate srv-cert
set security group-vpn server group grp1 ipsec-sa group-sa proposal group-prop
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 source 10.1.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 destination
10.2.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 source-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 protocol 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 source 10.2.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 destination
10.1.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 source-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 protocol 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 source 10.1.1.1/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 destination
239.1.1.1/32
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 source-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 protocol 0
set security group-vpn co-location
set security group-vpn member ipsec vpn v1 ike-gateway g1
set security group-vpn member ipsec vpn v1 group-vpn-external-interface ge-0/1/0
set security address-book book1 address 10_subnet 10.0.0.0/8
set security address-book book1 attach zone trust
set security address-book book2 address 10_subnet 10.0.0.0/8
set security address-book book2 attach zone untrust
set security policies from-zone trust to-zone untrust policy scope1 match source-address
10_subnet
set security policies from-zone trust to-zone untrust policy scope1 match destination-address
10_subnet
set security policies from-zone trust to-zone untrust policy scope1 match application any
set security policies from-zone trust to-zone untrust policy scope1 then permit tunnel ipsec-
group-vpn v1
set security policies from-zone untrust to-zone trust policy scope1 match source-address
10_subnet
set security policies from-zone untrust to-zone trust policy scope1 match destination-address
```

```
10_subnet
set security policies from-zone untrust to-zone trust policy scope1 match application any
set security policies from-zone untrust to-zone trust policy scope1 then permit tunnel ipsec-
group-vpn v1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure group VPN with server-member colocation:

1. Configure the loopback address on the device.

   ```
   [edit interfaces]
   user@host# set lo0 unit 0 family inet address 20.0.0.1/32
   ```

2. Configure the interface that member1 uses to connect to the MPLS network.

   ```
   [edit interfaces]
   user@host# set ge-0/1/0 unit 0 family inet address 10.1.0.1/32
   ```

3. Configure group VPN colocation on the device.

   ```
   [edit security group-vpn]
   user@host# set co-location
   ```

4. Configure IKE Phase 1 SA for the server (this configuration must match the Phase 1 SA configured on group members).

   ```
   [edit security group-vpn server ike proposal srv-prop]
   user@host# set authentication-method pre-shared-keys
   user@host# set dh-group group2
   user@host# set authentication-algorithm sha1
   user@host# set encryption-algorithm 3des-cbc
   ```

5. Define the IKE policy and set the remote gateways.

```
[edit security group-vpn server ike]
user@host# set policy srv-pol proposals srv-prop mode main pre-shared-key ascii-text
"$9$c1grK8-VYZUHX7UHqmF3Sre"
user@host# set gateway gw1 ike-policy srv-pol address 10.1.0.1
user@host# set gateway gw2 ike-policy srv-pol address 10.2.0.1
```

6. Configure the Phase 2 SA exchange for the server.

```
[edit security group-vpn server ipsec proposal group-prop]
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm 3des-cbc
user@host# set lifetime-seconds 3600
```

7. Configure the group identifier, IKE gateway, antireplay time, and server address on the server.

```
[edit security group-vpn server group grp1]
user@host# set group-id 1 anti-replay-time-window 120 server-address 20.0.0.1
user@host#set ike-gateway gw1
user@host#set ike-gateway gw2
```

8. Configure server to member communications.

```
[edit security group-vpn server group grp1]
user@host# set server-member-communication communication-type unicast encryption-algorithm
aes-128-cbc sig-hash-algorithm md5 certificate “srv-cert”
```

9. Configure the group policies to be downloaded to group members.

```
[edit security group-vpn server group grp1 ipsec-sa group-sa ]
user@host# set proposal group-prop match-policy p1 source 10.1.0.0/16 destination
10.2.0.0/16 source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p2 source 10.2.0.0/16 destination
10.1.0.0/16 source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p3 source 10.1.1.1/16 destination
239.1.1.1/32 source-port 0 destination-port 0 protocol 0
```

10. Configure Phase 1 SA for member1 (this configuration must match the Phase 1 SA configured for the group server).

```
[edit security group-vpn member ike proposal prop1]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm 3des-cbc
```

11. Define the policy and set the remote gateway for member1.

```
[edit security group-vpn member ike]
user@host# set policy pol1 mode main proposals prop1 pre-shared-key ascii-text "$9$c1grK8-
VYZUHX7UHqmF3Sre"
user@host# set gateway g1 ike-policy pol1 address 20.0.0.1 local-address 10.1.0.1
```

12. Configure the group identifier, IKE gateway, and interface for member1.

```
[edit security group-vpn member ipsec]
user@host# set vpn v1 group 1 ike-gateway g1 group-vpn-external-interface ge-0/1/0
```

13. Create address books and attach them to zones.

```
[edit security address-book book1]
user@member1# set address 10_subnet 10.0.0.0/8
user@member1# set attach zone trust
```

```
[edit security address-book book2]
user@member1# set address 10_subnet 10.0.0.0/8
user@member1# set attach zone untrust
```

14. Configure a scope policy from the trust zone to the untrust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone trust to-zone untrust]
user@member1# set policy scope1 match source-address 10_subnet destination-address
```

```
10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

15. Configure a scope policy from the untrust zone to the trust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone untrust to-zone trust]
user@member1# set policy scope1 match source-address 10_subnet destination-address
10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

Results

From configuration mode, confirm your configuration by entering the `show security group-vpn` and `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

In the list of configured security policies, make sure that the scope policies are listed before the default policies.

```
[edit]
user@host# show security group-vpn
member {
    ike {
        proposal prop1 {
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
        }
        policy pol1 {
            mode main;
            proposals prop1;
            pre-shared-key ascii-text "$9$c1grK8-VYZUHX7UHqmF3Sre"; ## SECRET-DATA
        }
        gateway g1 {
            ike-policy pol1;
            address 20.0.0.1;
            local-address 10.1.0.1;
        }
    }
```

```
ipsec {
    vpn v1 {
        ike-gateway g1;
        group-vpn-external-interface ge-0/1/0;
        group 1;
    }
}
}
server {
    ike {
        proposal srv-prop {
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
        }
        policy srv-pol {
            mode main;
            proposals srv-prop;
            pre-shared-key ascii-text "$9$c1grK8-VYZUHX7UHqmF3Sre"; ## SECRET-DATA
        }
        gateway gw1 {
            ike-policy srv-pol;
            address 10.1.0.1;
        }
        gateway gw2 {
            ike-policy srv-pol;
            address 10.2.0.1;
        }
    }
    ipsec {
        proposal group-prop {
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 3600;
        }
    }
    group grp1 {
        group-id 1;
        ike-gateway gw1;
        ike-gateway gw2;
        anti-replay-time-window 120;
        server-address 20.0.0.1;
```

```
            server-member-communication {
                communication-type unicast;
                encryption-algorithm aes-128-cbc;
                sig-hash-algorithm md5;
                certificate srv-cert;
            }
            ipsec-sa group-sa {
                proposal group-prop;
                match-policy p1 {
                    source 10.1.0.0/16;
                    destination 10.2.0.0/16;
                    source-port 0;
                    destination-port 0;
                    protocol 0;
                }
                match-policy p2 {
                    source 10.2.0.0/16;
                    destination 10.1.0.0/16;
                    source-port 0;
                    destination-port 0;
                    protocol 0;
                }
                match-policy p3 {
                    source 10.1.1.1/16;
                    destination 239.1.1.1/32;
                    source-port 0;
                    destination-port 0;
                    protocol 0;
                }
            }
        }
    }
co-location;
```

```
[edit]
user@host# show security policies
from-zone trust to-zone trust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone trust to-zone untrust {
    policy scope1 {
        match {
            source-address 10_subnet;
            destination-address 10_subnet;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-group-vpn v1;
                }
            }
        }
    }
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone untrust to-zone trust {
    policy default-deny {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            deny;
        }
    }
    policy scope1 {
        match {
            source-address 10_subnet;
            destination-address 10_subnet;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-group-vpn v1;
                }
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying Group VPN Member Registration

Purpose

Verify that the group VPN members are registered correctly.

Action

From operational mode, enter the `show security group-vpn registered-members` command.

Verifying Group VPN Server Security Associations for IKE

Purpose

Verify the SAs for the group VPN server for IKE.

Action

From operational mode, enter the `show security group-vpn server ike security-associations` command.

Verifying Group VPN Server Security Associations for IPsec

Purpose

Verify the SAs for the group VPN server for IPsec.

Action

From operational mode, enter the `show security group-vpn server ipsec security-associations` command.

Verifying Group VPN Member Security Associations for IKE

Purpose

Verify the SAs for the group VPN members for IKE.

Action

From operational mode, enter the `show security group-vpn member ike security-associations` command.

Verifying Group VPN Member Security Associations for IPsec

Purpose

Verify the SAs for the group VPN members for IPsec.

Action

From operational mode, enter the `show security group-vpn member ipsec security-associations` command.

Release History Table

Release	Description
12.3X48-D30	Starting with Junos OS Release 12.3X48-D30, Group VPNv1 members can interoperate with Group VPNv2 servers.
12.3X48-D30	Starting with Junos OS Release 12.3X48-D30, Group VPNv1 members on SRX100, SRX110, SRX210, SRX220, SRX240, SRX550, and SRX650 devices can interoperate with Group VPNv2 servers.

RELATED DOCUMENTATION

Monitoring VPN Traffic | 1351

Group VPNv2

IN THIS SECTION

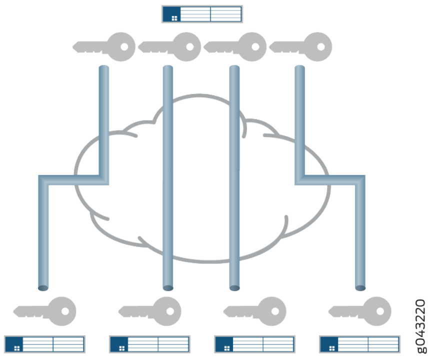
Group VPNv2 introduces the concept of a trusted group to eliminate point-to-point tunnels and their associated overlay routing. All group members share a common security association (SA), also known as a group SA.

Group VPNv2 Overview

IN THIS SECTION

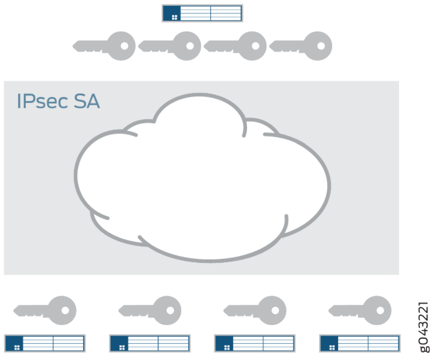
An IPsec security association (SA) is a unidirectional agreement between virtual private network (VPN) participants that defines the rules to use for authentication and encryption algorithms, key exchange mechanisms, and secure communications. With many VPN implementations, the SA is a point-to-point tunnel between two security devices (see Figure 52 on page 758).

Figure 52: Point-to-Point SAs



Group VPNv2 extends IPsec architecture to support SAs that are shared by a group of security devices (see Figure 53 on page 759). With Group VPNv2, any-to-any connectivity is achieved by preserving the original source and destination IP addresses in the outer header. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

Figure 53: Shared SAs



Group VPNv2 is an enhanced version of the group VPN feature introduced in an earlier Junos OS release for SRX Series Firewalls. Group VPNv2 on Juniper devices support RFC 6407, *The Group Domain of Interpretation (GDOI)*, and interoperate with other devices that comply with RFC 6407.

Understanding the GDOI Protocol for Group VPNv2

Group VPNv2 is based on RFC 6407, *The Group Domain of Interpretation* (GDOI). This RFC describes the protocol between group members and group servers to establish SAs among group members. GDOI messages create, maintain, or delete SAs for a group of devices. Group VPNv2 is supported on vSRX Virtual Firewall instances and all SRX Series Firewalls except for SRX5400, SRX5600, and SRX5800 devices.

The GDOI protocol runs on UDP port 848. The Internet Security Association and Key Management Protocol (ISAKMP) defines two negotiation phases to establish SAs for an IKE IPsec tunnel. Phase 1 allows two devices to establish an ISAKMP SA for other security protocols, such as GDOI.

With Group VPNv2, Phase 1 ISAKMP SA negotiation is performed between a group server and a group member. The server and member must use the same ISAKMP policy. GDOI exchanges between the server and member establish the SAs that are shared with other group members. A group member does not need to negotiate IPsec with other group members. GDOI exchanges must be protected by ISAKMP Phase 1 SAs.

There are two types of GDOI exchanges:

- The `groupkey-pull` exchange allows a member to request SAs and keys shared by the group from the server. Group members must register with a group server through a `groupkey-pull` exchange.

- The `groupkey-push` exchange is a single rekey message that allows the server to send group SAs and keys to members before existing group SAs expire. Rekey messages are unsolicited messages sent from the server to members.

Understanding Group VPNv2 Servers and Members

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. The center of Group VPNv2 is the group controller/key server (GCKS). A server cluster can be used to provide GCKS redundancy.

The GCKS or group server performs the following tasks:

- Controls group membership.

- Generates encryption keys.

- Sends new group SAs and keys to members. Group members encrypt traffic based on the group SAs and keys provided by the group server.

A group server can service multiple groups. A single security device can be a member of multiple groups.

Each group is represented by a group identifier, which is a number between 1 and 4,294,967,295. The group server and group members are linked together by the group identifier. There can be only one group identifier per group, and multiple groups cannot use the same group identifier.

The following is a high-level view of Group VPNv2 server and member actions:

1. The group server listens on UDP port 848 for members to register.

2. To register with the group server, the member first establishes an IKE SA with the server. A member device must provide correct IKE Phase 1 authentication to join the group. Preshared key authentication on a per-member basis is supported.

3. Upon successful authentication and registration, the member device retrieves group SAs and keys for the specified group identifier from the server with a GDOI `groupkey-pull` exchange.

4. The server adds the member to the membership for the group.

5. Group members exchange packets encrypted with group SA keys.

The server sends SA and key refreshes to group members with rekey (GDOI `groupkey-push`) messages. The server sends rekey messages before SAs expire to ensure that valid keys are available for encrypting traffic between group members.

A rekey message sent by the server requires an acknowledgement (ack) message from each group member. If the server does not receive an ack message from the member, the rekey message is retransmitted at the configured `retransmission-period` (the default is 10 seconds). If there is no reply from the member after the configured `number-of-retransmission` (the default is 2 times), the member is removed from the server's registered members. The IKE SA between the server and member is also removed.

The server also sends rekey messages to provide new keys to members when the group SA has changed.

Understanding Group VPNv2 Limitations

Group VPNv2 servers only operate with Group VPNv2 members that support RFC 6407, *The Group Domain of Interpretation (GDOI)*.

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. The following are not supported in this release for Group VPNv2:

- SNMP.
- Deny policy from Cisco GET VPN server.
- PKI support for Phase 1 IKE authentication.
- Colocation of group server and member, where server and member functions coexist in the same physical device.
- Group members configured as chassis clusters.
- J-Web interface for configuration and monitoring.
- Multicast data traffic.

Group VPNv2 is not supported in deployments where IP addresses cannot be preserved—for example, across the Internet where NAT is used.

Understanding Group VPNv2 Server-Member Communication

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. Server-member communication allows the server to send GDOI `groupkey-push` (rekey) messages to members. If server-member communication is not configured for the group, members can send GDOI `groupkey-pull` messages to register and reregister with the server, but the server is not able to send `groupkey-push` messages to members.

Server-member communication is configured for the group by using the `server-member-communication` *configuration statement* at the [`edit security group-vpn server`] hierarchy. The following options can be defined:

- Authentication algorithm (sha-256 or sha-384) used to authenticate the member to the server. There is no default algorithm.

- Encryption algorithm used for communications between the server and member. You can specify aes-128-cbc, aes-192-cbc, or aes-256-cbc. There is no default algorithm.

- Unicast communication type for rekey messages sent to group members.

- Lifetime for the key encryption key (KEK). The default is 3600 seconds.

- Number of times the group server retransmits `groupkey-push` messages to a group member without a response (the default is 2 times) and the period of time between retransmissions (the default is 10 seconds).

If server-member communication for a group is not configured, the membership list displayed by the `show security group-vpn server registered-members` command shows group members who have registered with the server; members can be active or not. When server-member communication for a group is configured, the group membership list is cleared. For unicast communication type, the `show security group-vpn server registered-members` command shows only active members.

Understanding Group VPNv2 Key Operations

This topic contains the following sections:

Group Keys

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. The group server maintains a database to track the relationship among VPN groups, group members, and group keys. There are two kinds of group keys that the server downloads to members:

- Key Encryption Key (KEK)—Used to encrypt SA rekey (GDOI `groupkey-push`) exchanges. One KEK is supported per group.

- Traffic Encryption Key (TEK)—Used to encrypt and decrypt IPsec data traffic between group members.

The key associated with an SA is accepted by a group member only if there is a matching policy configured on the member. An accepted key is installed for the group, whereas a rejected key is discarded.

Rekey Messages

If the group is configured for server-member communications, the server sends SA and key refreshes to group members with rekey (GDOI `groupkey-push`) messages. Rekey messages are sent before SAs expire; this ensures that valid keys are available for encrypting traffic between group members.

The server also sends rekey messages to provide new keys to members when there is a change in group membership or the group SA has changed (for example, a group policy is added or deleted).

Server-member communications options must be configured on the server to allow the server to send rekey messages to group members.

The group server sends one copy of the unicast rekey message to each group member. Upon receipt of the rekey message, members must send an acknowledgment (ACK) to the server. If the server does not receive an ACK from a member (including retransmission of rekey messages), the server considers the member to be inactive and removes it from the membership list. The server stops sending rekey messages to the member.

The `number-of-retransmission` and `retransmission-period` configuration statements for server-member communications control the resending of rekey messages by the server when no ACK is received from a member.

The interval at which the server sends rekey messages is based on the value of the `lifetime-seconds` configuration statement at the [`edit security group-vpn server group` *`group-name`*] hierarchy. New keys are generated before the expiration of the KEK and TEK keys.

The `lifetime-seconds` for the KEK is configured as part of the server-member communications; the default is 3600 seconds. The `lifetime-seconds` for the TEK is configured for the IPsec proposal; the default is 3600 seconds.

Member Registration

If a group member does not receive a new SA key from the server before the current key expires, the member must reregister with the server and obtain updated keys with a GDOI `groupkey-pull` exchange.

Group VPNv2 Configuration Overview

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. This topic describes the main tasks for configuring Group VPNv2.

The group controller/key server (GCKS) manages Group VPNv2 security associations (SAs), and generates encryption keys and distributes them to group members. You can use a Group VPNv2 server cluster to provide GCKS redundancy. See "Understanding Group VPNv2 Server Clusters" on page 818.

On the group server(s), configure the following:

1. IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv2 " on page 765.
2. IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv2" on page 765.
3. VPN group information, including the group identifier, IKE gateways for group members, the maximum number of members in the group, and server-member communications. Group configuration includes a group policy that defines the traffic to which the SA and keys apply. Server cluster and antireplay time window can optionally be configured. See "Group VPNv2 Configuration Overview" on page 763 and "Understanding Group VPNv2 Traffic Steering" on page 766.

On the group member, configure the following:

1. IKE Phase 1 SA. See "Understanding IKE Phase 1 Configuration for Group VPNv2 " on page 765.

2. IPsec SA. See "Understanding IPsec SA Configuration for Group VPNv2" on page 765.

3. IPsec policy that defines the incoming zone (usually a protected LAN), outgoing zone (usually a WAN) and the VPN group to which the policy applies. Exclude or fail-open rules can also be specified. See "Understanding Group VPNv2 Traffic Steering" on page 766.

4. Security policy to allow group VPN traffic between the zones specified in the IPsec policy.

Group VPNv2 operation requires a working routing topology that allows client devices to reach their intended sites throughout the network.

The group is configured on the server with the `group` *configuration statement* at the [`edit security group-vpn server`] hierarchy.

The group information consists of the following information:

- Group identifier—A value that identifies the VPN group. The same group identifier must be configured on the group member.

- Each group member is configured with the `ike-gateway` configuration statement. There can be multiple instances of this configuration statement, one for each member of the group.

- Group policies—Policies that are to be downloaded to members. Group policies describe the traffic to which the SA and keys apply. See "Understanding Group VPNv2 Traffic Steering" on page 766.

- Member threshold—The maximum number of members in the group. After the member threshold for a group is reached, a server stops responding to `groupkey-pull` initiations from new members. See "Understanding Group VPNv2 Server Clusters" on page 818.

- Server-member communication—Optional configuration that allows the server to send `groupkey-push` rekey messages to members.

- Server cluster—Optional configuration that supports group controller/key server (GCKS) redundancy. See "Understanding Group VPNv2 Server Clusters" on page 818.

- Antireplay—Optional configuration that detects packet interception and replay. See "Understanding Group VPNv2 Antireplay" on page 769.

Understanding IKE Phase 1 Configuration for Group VPNv2

An IKE Phase 1 SA between a group server and a group member establishes a secure channel in which to negotiate IPsec SAs that are shared by a group. For standard IPsec VPNs on Juniper Networks security devices, Phase 1 SA configuration consists of specifying an IKE proposal, policy, and gateway.

For Group VPNv2, the IKE Phase 1 SA configuration is similar to the configuration for standard IPsec VPNs, but is performed at the [`edit security group-vpn server ike`] and [`edit security group-vpn member ike`] hierarchies. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

In the IKE proposal configuration, you set the authentication method and the authentication and encryption algorithms that will be used to open a secure channel between participants. In the IKE policy configuration, you set the mode in which the Phase 1 channel will be negotiated, specify the type of key exchange to be used, and reference the Phase 1 proposal. In the IKE gateway configuration, you reference the Phase 1 policy.

The IKE proposal and policy configuration on the group server must match the IKE proposal and policy configuration on group members. On a group server, an IKE gateway is configured for each group member. On a group member, up to four server addresses can be specified in the IKE gateway configuration.

Understanding IPsec SA Configuration for Group VPNv2

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. After the server and member have established a secure and authenticated channel in Phase 1 negotiation, they proceed to establish the IPsec SAs that are shared by group members to secure data that is transmitted among members. While the IPsec SA configuration for Group VPNv2 is similar to the configuration for standard VPNs, a group member does not need to negotiate the SA with other group members.

IPsec configuration for Group VPNv2 consists of the following information:

- On the group server, an IPsec proposal is configured for the security protocol, authentication, and encryption algorithm to be used for the SA. The IPsec SA proposal is configured on the group server with the `proposal` configuration statement at the [`edit security group-vpn server ipsec`] hierarchy.
- On the group member, an Autokey IKE is configured that references the group identifier, the group server (configured with the `ike-gateway` configuration statement), and the interface used by the member to connect to group peers. The Autokey IKE is configured on the member with the `vpn` configuration statement at the [`edit security group-vpn member ipsec`] hierarchy.

SEE ALSO

Understanding Group VPNv2 Server Clusters | 818

Understanding Group VPNv2 Traffic Steering

IN THIS SECTION

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. The group server distributes IPsec security associations (SAs) and keys to members of a specified group. All members that belong to the same group share the same set of IPsec SAs. The SA that is installed on a specific group member is determined by the policy associated with the group SA and the IPsec policy that is configured on the group member.

Group Policies Configured on Group Servers

In a VPN group, each group SA and key that the server pushes to a member are associated with a group policy. The group policy describes the traffic on which the key should be used, including protocol, source address, source port, destination address, and destination port. On the server, the group policy is configured with the `match-policy` *`policy-name`* options at the [`edit security group-vpn server group` *`name`* `ipsec-sa` *`name`*] hierarchy level.

Group policies that are identical (configured with the same source address, destination address, source port, destination port, and protocol values) cannot exist for a single group. An error is returned if you attempt to commit a configuration that contains identical group policies for a group. If this occurs, you must delete one of the identical group policies before you can commit the configuration.

IPsec Policies Configured on Group Members

On the group member, an IPsec policy consists of the following information:

- Incoming zone (`from-zone`) for group traffic.

- Outgoing zone (`to-zone`) for group traffic.

- The name of the group to which the IPsec policy applies. Only one Group VPNv2 name can be referenced by a specific from-zone/to-zone pair.

The interface that is used by the group member to connect to the Group VPNv2 must belong to the outgoing zone. This interface is specified with the `group-vpn-external-interface` statement at the [`edit security group-vpn member ipsec vpn` *`vpn-name`*] hierarchy level.

On the group member, the IPsec policy is configured at the [`edit security ipsec-policy`] hierarchy level. Traffic that matches the IPsec policy is further checked against exclude and fail-open rules that are configured for the group.

Fail-Close

By default, traffic that does not match exclude or fail-open rules or group policies received from the group server is blocked; this is known as *fail-close*.

Exclude and Fail-Open Rules

On group members, the following types of rules can be configured for each group:

- Traffic that is excluded from VPN encryption. Examples of this type of traffic can include BGP or OSPF routing protocols. To exclude traffic from a group, use the `set security group-vpn member ipsec vpn` *`vpn-name`* `exclude rule` configuration. A maximum of 10 exclude rules can be configured.

- Traffic that is critical to the customer's operation and must be sent in cleartext (unencrypted) if the group member has not received a valid traffic encryption key (TEK) for the IPsec SA. Fail-open rules allow this traffic flow while all other traffic is blocked. Enable fail-open with the `set security group-vpn member ipsec vpn` *`vpn-name`* `fail-open rule` configuration. A maximum of 10 fail-open rules can be configured.

Priorities of IPsec Policies and Rules

IPsec policies and rules have the following priorities on the group member:

1. Exclude rules that define traffic to be excluded from VPN encryption.

2. Group policies that are downloaded from the group server.

3. Fail-open rules that define traffic that is sent in cleartext if there is no valid TEK for the SA.

4. Fail-close policy that blocks traffic. This is the default if traffic does not match exclude or fail-open rules or group policies.

SEE ALSO

Understanding Configuration Changes with Group VPNv2 Server Clusters | **826**

Understanding the Group VPNv2 Recovery Probe Process

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. Two situations could indicate that a group member is out of synchronization with the group server and other group members:

- The group member receives an Encapsulating Security Payload (ESP) packet with an unrecognized Security Parameter Index (SPI).

- There is outgoing IPsec traffic but no incoming IPsec traffic on the group member.

When either situation is detected, a recovery probe process can be triggered on the group member. The recovery probe process initiates GDOI `groupkey-pull` exchanges at specific intervals to update the member's SA from the group server. If there is a DoS attack of bad SPI packets or if the sender itself is out of synchronization, the out-of-synchronization indication on the group member might be a false alarm. To avoid overloading the system, the `groupkey-pull` initiation is retried at intervals of 10, 20, 40, 80, 160, and 320 seconds.

The recovery probe process is disabled by default. To enable the recovery probe process, configure `recovery-probe` at the [`edit security group-vpn member ipsec vpn` *`vpn-name`*] hierarchy level.

Understanding Group VPNv2 Antireplay

Group VPNv2 antireplay is supported on vSRX Virtual Firewall instances and all SRX Series Firewalls except for SRX5400, SRX5600, and SRX5800 devices. Antireplay is an IPsec feature that can detect when a packet is intercepted and then replayed by attackers. Antireplay is disabled by default for a group.

Each IPsec packet contains a timestamp. The group member checks whether the packet's timestamp falls within the configured `anti-replay-time-window` value. A packet is dropped if the timestamp exceeds the value.

We recommend that NTP be configured on all devices that support Group VPNv2 antireplay.

Group members that are running on vSRX Virtual Firewall instances on a host machine where the hypervisor is running under a heavy load can experience issues that can be corrected by reconfiguring the `anti-replay-time-window` value. If data that matches the IPsec policy on the group member is not being transferred, check the `show security group-vpn member ipsec statistics` output for D3P errors. Make sure that NTP is operating correctly. If there are errors, adjust the `anti-replay-time-window` value.

SEE ALSO

Understanding Antireplay for Group VPNv1 | 716

Example: Configuring a Group VPNv2 Server and Members

IN THIS SECTION

This example shows how to configure a Group VPNv2 server to provide group controller/key server (GCKS) support to Group VPNv2 group members. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

Requirements

The example uses the following hardware and software components:

- A supported SRX Series Firewall or vSRX Virtual Firewall instance running Junos OS Release 15.1X49-D30 or later that supports Group VPNv2. This SRX Series Firewall or vSRX Virtual Firewall instance operates as a Group VPNv2 server.
- Two supported SRX Series Firewalls or vSRX Virtual Firewall instances running Junos OS Release 15.1X49-D30 or later that support Group VPNv2. These devices or instances operate as Group VPNv2 group members.
- Two supported MX Series devices running Junos OS Release 15.1R2 or later that support Group VPNv2. These devices operate as Group VPNv2 group members.

A hostname, a root administrator password, and management access must be configured on each device. We recommend that NTP also be configured on each device.

Group VPNv2 operation requires a working routing topology that allows client devices to reach their intended sites throughout the network. This examples focuses on the Group VPNv2 configuration; the routing configuration is not described.

Overview

IN THIS SECTION

In this example, the Group VPNv2 network consists of a server and four members. Two of the members are SRX Series Firewalls or vSRX Virtual Firewall instances while the other two members are MX Series devices. The shared group VPN SAs secure traffic between group members.

The group VPN SAs must be protected by a Phase 1 SA. Therefore, the group VPN configuration must include configuring IKE Phase 1 negotiations on both the group server and the group members.

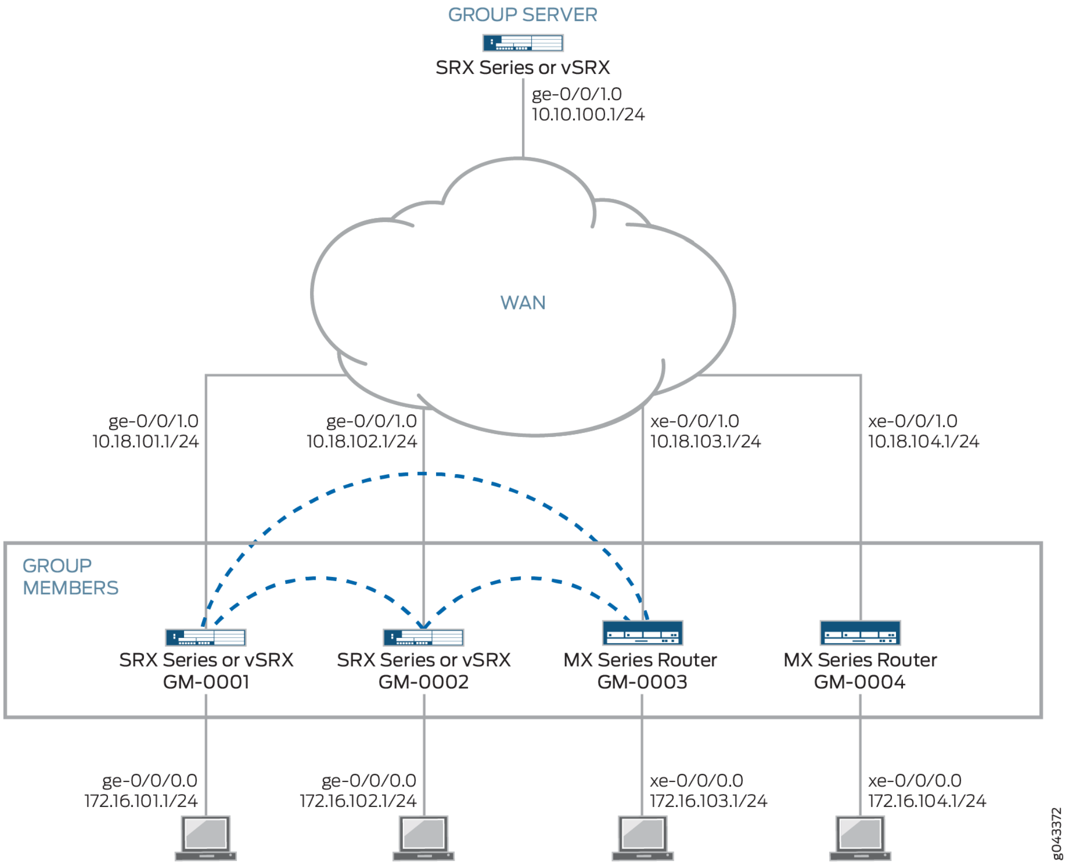
The same group identifier must be configured on both the group server and the group members. In this example, the group name is GROUP_ID-0001 and the group identifier is 1. The group policy configured on the server specifies that the SA and key are applied to traffic between subnetworks in the 172.16.0.0/12 range.

On SRX Series Firewall or vSRX Virtual Firewall group members, an IPsec policy is configured for the group with the LAN zone as the from-zone (incoming traffic) and the WAN zone as the to-zone (outgoing traffic). A security policy is also needed to allow traffic between the LAN and WAN zones.

Topology

Figure 54 on page 771 shows the Juniper Networks devices to be configured for this example.

Figure 54: Group VPNv2 Server with SRX Series Firewall or vSRX Virtual Firewall and MX Series Members



Configuration

IN THIS SECTION

Configuring the Group Server

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.10.100.1/24
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then reject
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set routing-options static route 10.18.101.0/24 next-hop 10.10.100.254
set routing-options static route 10.18.102.0/24 next-hop 10.10.100.254
set routing-options static route 10.18.103.0/24 next-hop 10.10.100.254
set routing-options static route 10.18.104.0/24 next-hop 10.10.100.254
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.10.100.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.10.100.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.10.100.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
set security group-vpn server ike gateway GM-0004 local-address 10.10.100.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-
sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0005
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type
unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-
algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds
7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm
sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-
L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source
172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1
destination 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol
0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 server:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.10.100.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0
[edit security policies]
user@host# set global policy 1000 match source-address any
user@host# set global policy 1000 match destination-address any
user@host# set global policy 1000 match application any
user@host# set global policy 1000 match from-zone any
user@host# set global policy 1000 match to-zone any
user@host# set global policy 1000 then reject
user@host# set global policy 1000 then log session-init
user@host# set global policy 1000 then count
user@host# set default-policy deny-all
```

2. Configure the static routes.

```
[edit routing-options]
user@host# set static route 10.18.101.0/24 next-hop 10.10.100.254
user@host# set static route 10.18.102.0/24 next-hop 10.10.100.254
user@host# set static route 10.18.103.0/24 next-hop 10.10.100.254
user@host# set static route 10.18.104.0/24 next-hop 10.10.100.254
```

3. Configure the IKE proposal, policy, and gateways.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set authentication-algorithm sha-256
user@host# set dh-group group14
```

```
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy  GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike gateway GM-0001]
user@host# set ike-policy GMs
user@host# set address 10.18.101.1
user@host# set local-address 10.10.100.1
[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.10.100.1
[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.10.100.1
[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.10.100.1
```

4. Configure the IPsec proposal.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600 VPN Group
```

5. Configure the group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
```

6. Configure server-to-member communications.

```
[edit security group-vpn server group GROUP_ID-0001 server-member-communication]
user@host# set communication-type unicast
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 7200
user@host# set sig-hash-algorithm sha-256
```

7. Configure the group policy to be downloaded to the group members.

```
[edit security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001]
user@host# set proposal AES256-SHA256-L3600
user@host# set match-policy 1 source 172.16.0.0/12
user@host# set match-policy 1 destination 172.16.0.0/12
user@host# set match-policy 1 protocol 0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.10.100.1/24;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 10.18.101.0/24 next-hop 10.10.100.254;
    route 10.18.102.0/24 next-hop 10.10.100.254;
    route 10.18.103.0/24 next-hop 10.10.100.254;
    route 10.18.104.0/24 next-hop 10.10.100.254;
}
```

```
[edit]
user@host# show security
group-vpn {
    server {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
                encryption-algorithm aes-256-cbc;
            }
            policy GMs {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            gateway GM-0001 {
                ike-policy GMs;
                address 10.18.101.1;
                local-address 10.10.100.1;
            }
            gateway GM-0002 {
                ike-policy GMs;
                address 10.18.102.1;
                local-address 10.10.100.1;
            }
            gateway GM-0003 {
                ike-policy GMs;
                address 10.18.103.1;
                local-address 10.10.100.1;
            }
            gateway GM-0004 {
                ike-policy GMs;
                address 10.18.104.1;
                local-address 10.10.100.1;
            }
        }
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
                lifetime-seconds 3600;
            }
        }
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            ike-gateway GM-0001;
            ike-gateway GM-0002;
            ike-gateway GM-0003;
            ike-gateway GM-0004;
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                reject;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Group Member GM-0001 (SRX Series Firewall or vSRX Virtual Firewall Instance)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_LAN
set interfaces ge-0/0/0 unit 0 family inet address 172.16.101.1/24
set interfaces ge-0/0/1 unit 0 description To_KeySrv
set interfaces ge-0/0/1 unit 0 family inet address 10.18.101.1/24
set security zones security-zone LAN host-inbound-traffic system-services ike
set security zones security-zone LAN host-inbound-traffic system-services ssh
set security zones security-zone LAN host-inbound-traffic system-services ping
set security zones security-zone LAN interfaces ge-0/0/0.0
set security zones security-zone WAN host-inbound-traffic system-services ike
set security zones security-zone WAN host-inbound-traffic system-services ssh
set security zones security-zone WAN host-inbound-traffic system-services ping
set security zones security-zone WAN interfaces ge-0/0/1.0
set security address-book global address 172.16.0.0/12 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match application any
set security policies from-zone LAN to-zone WAN policy 1 then permit
set security policies from-zone LAN to-zone WAN policy 1 then log session-init
set security policies from-zone WAN to-zone LAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match application any
set security policies from-zone WAN to-zone LAN policy 1 then permit
set security policies from-zone WAN to-zone LAN policy 1 then log session-init
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then reject
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set routing-options static route 10.18.102.0/24 next-hop 10.18.101.254
set routing-options static route 10.18.103.0/24 next-hop 10.18.101.254
set routing-options static route 10.18.104.0/24 next-hop 10.18.101.254
set routing-options static route 172.16.101.0/24 next-hop 10.18.101.254
set routing-options static route 172.16.102.0/24 next-hop 10.18.101.254
set routing-options static route 172.16.103.0/24 next-hop 10.18.101.254
set routing-options static route 172.16.104.0/24 next-hop 10.18.101.254
set routing-options static route 10.10.100.0/24 next-hop 10.18.101.254
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy KeySrv mode main
set security group-vpn member ike policy KeySrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy KeySrv pre-shared-key ascii-text "$ABC123"
set security group-vpn member ike gateway KeySrv ike-policy KeySrv
set security group-vpn member ike gateway KeySrv server-address 10.10.100.1
set security group-vpn member ike gateway KeySrv local-address 10.18.101.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway KeySrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group-vpn-external-interface ge-0/0/1.0
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 recovery-probe
set security ipsec-policy from-zone LAN to-zone WAN ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_LAN
user@host# set ge-0/0/0 unit 0 family inet address 172.16.101.1/24
user@host# set ge-0/0/1 unit 0 description To_KeySrv
user@host# set ge-0/0/1 unit 0 family inet address 10.18.101.1/24
[edit security zones security-zone LAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
[edit security]
user@host# set address-book global address 172.16.0.0/12 172.16.0.0/12
[edit security zones security-zone WAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0
[edit security policies from-zone LAN to-zone WAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set then log session-init
[edit security policies from-zone WAN to-zone LAN
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
```

```
user@host# set then log session-init
[edit security policies]
user@host# set global policy 1000 match source-address any
user@host# set global policy 1000 match destination-address any
user@host# set global policy 1000 match application any
user@host# set global policy 1000 match from-zone any
user@host# set global policy 1000 match to-zone any
user@host# set global policy 1000 match then reject
user@host# set global policy 1000 match then log session-init
user@host# set global policy 1000 match then count
user@host# set default-policy deny-all
```

2. Configure the static routes.

```
[edit routing-options]
user@host# set static route 10.18.102.0/24 next-hop 10.18.101.254
user@host# set static route 10.18.103.0/24 next-hop 10.18.101.254
user@host# set static route 10.18.104.0/24 next-hop 10.18.101.254
user@host# set static route 172.16.101.0/24 next-hop 10.18.101.254
user@host# set static route 172.16.102.0/24 next-hop 10.18.101.254
user@host# set static route 172.16.103.0/24 next-hop 10.18.101.254
user@host# set static route 172.16.104.0/24 next-hop 10.18.101.254
user@host# set static route 10.10.100.0/24 next-hop 10.18.101.254
```

3. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set authentication-algorithm sha-256
user@host# set dh-group group14
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy  KeySrv ]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn member ike gateway KeySrv]
user@host# set ike-policy KeySrv
user@host# set server-address 10.10.100.1
user@host# set local-address 10.18.101.1
```

4. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway KeySrv
user@host# set group-vpn-external-interface ge-0/0/1.0
user@host# set group 1
user@host# set recovery-probe
```

5. Configure the IPsec policy.

```
[edit security ipsec-policy from-zone LAN to-zone WAN]
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_LAN;
        family inet {
            address 172.16.101.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_KeySrv;
        family inet {
            address 10.18.101.1/24;
        }
    }
}
[edit]
user@host# show routing-options
```

```
static {
    route 10.18.102.0/24 next-hop 10.18.101.254;
    route 10.18.103.0/24 next-hop 10.18.101.254;
    route 10.18.104.0/24 next-hop 10.18.101.254;
    route 172.16.101.0/24 next-hop 10.18.101.254;
    route 172.16.102.0/24 next-hop 10.18.101.254;
    route 172.16.103.0/24 next-hop 10.18.101.254;
    route 172.16.104.0/24 next-hop 10.18.101.254;
    route 10.10.100.0/24 next-hop 10.18.101.254;
}
[edit]
user@host# show security
address-book {
    global {
        address 172.16.0.0/12 172.16.0.0/12;
    }
}
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy KeySrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            gateway KeySrv {
                ike-policy KeySrv;
                server-address 10.10.100.1;
                local-address 10.18.101.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway KeySrv;
                group-vpn-external-interface ge-0/0/1.0;
                group 1;
                recovery-probe;
            }
        }
    }
}
ipsec-policy {
    from-zone LAN to-zone WAN {
        ipsec-group-vpn GROUP_ID-0001;
    }
}
policies {
    from-zone LAN to-zone WAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    from-zone WAN to-zone LAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                reject;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone LAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone WAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Group Member GM-0002 (SRX Series Firewall or vSRX Virtual Firewall Instance)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_LAN
set interfaces ge-0/0/0 unit 0 family inet address 172.16.102.1/24
set interfaces ge-0/0/1 unit 0 description To_KeySrv
set interfaces ge-0/0/1 unit 0 family inet address 10.18.102.1/24
set security zones security-zone LAN host-inbound-traffic system-services ike
set security zones security-zone LAN host-inbound-traffic system-services ssh
set security zones security-zone LAN host-inbound-traffic system-services ping
set security zones security-zone LAN interfaces ge-0/0/0.0
set security zones security-zone WAN host-inbound-traffic system-services ike
set security zones security-zone WAN host-inbound-traffic system-services ssh
set security zones security-zone WAN host-inbound-traffic system-services ping
set security zones security-zone WAN interfaces ge-0/0/1.0
set security address-book global address 172.16.0.0/12 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match application any
set security policies from-zone LAN to-zone WAN policy 1 then permit
set security policies from-zone LAN to-zone WAN policy 1 then log session-init
set security policies from-zone WAN to-zone LAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match application any
set security policies from-zone WAN to-zone LAN policy 1 then permit
set security policies from-zone WAN to-zone LAN policy 1 then log session-init
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then reject
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set routing-options static route 10.18.101.0/24 next-hop 10.18.102.254
set routing-options static route 10.18.103.0/24 next-hop 10.18.102.254
set routing-options static route 10.18.104.0/24 next-hop 10.18.102.254
set routing-options static route 172.16.101.0/24 next-hop 10.18.102.254
set routing-options static route 172.16.102.0/24 next-hop 10.18.102.254
set routing-options static route 172.16.103.0/24 next-hop 10.18.102.254
set routing-options static route 172.16.104.0/24 next-hop 10.18.102.254
set routing-options static route 10.10.100.0/24 next-hop 10.18.102.254
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy KeySrv mode main
set security group-vpn member ike policy KeySrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy KeySrv pre-shared-key ascii-text "$ABC123"
set security group-vpn member ike gateway KeySrv ike-policy KeySrv
set security group-vpn member ike gateway KeySrv server-address 10.10.100.1
set security group-vpn member ike gateway KeySrv local-address 10.18.102.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway KeySrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group-vpn-external-interface ge-0/0/1.0
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 recovery-probe
set security ipsec-policy from-zone LAN to-zone WAN ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_LAN
user@host# set ge-0/0/0 unit 0 family inet address 172.16.102.1/24
```

```
user@host# set ge-0/0/1 unit 0 description To_KeySrv
user@host# set ge-0/0/1 unit 0 family inet address 10.18.101.1/24
[edit security zones security-zone LAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
[edit security zones security-zone WAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0
[edit security]
user@host# set address-book global address 172.16.0.0/12 172.16.0.0/12
[edit security policies from-zone LAN to-zone WAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set then log session-init
[edit security policies from-zone WAN to-zone LAN
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set then log session-init
[edit security policies]
user@host# set global policy 1000 match source-address any
user@host# set global policy 1000 match destination-address any
user@host# set global policy 1000 match application any
user@host# set global policy 1000 match from-zone any
user@host# set global policy 1000 match to-zone any
user@host# set global policy 1000 match then reject
user@host# set global policy 1000 match then log session-init
user@host# set global policy 1000 match then count
user@host# set default-policy deny-all
```

2. Configure the static routes.

```
[edit routing-options]
user@host# set static route 10.18.101.0/24 next-hop 10.18.102.254
user@host# set static route 10.18.103.0/24 next-hop 10.18.102.254
```

```
user@host# set static route 10.18.104.0/24 next-hop 10.18.102.254
user@host# set static route 172.16.101.0/24 next-hop 10.18.102.254
user@host# set static route 172.16.102.0/24 next-hop 10.18.102.254
user@host# set static route 172.16.103.0/24 next-hop 10.18.102.254
user@host# set static route 172.16.104.0/24 next-hop 10.18.102.254
user@host# set static route 10.10.100.0/24 next-hop 10.18.102.254
```

3. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set authentication-algorithm sha-256
user@host# set dh-group group14
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy  KeySrv ]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn member ike gateway KeySrv]
user@host# set ike-policy KeySrv
user@host# set server-address 10.10.100.1
user@host# set local-address 10.18.102.1
```

4. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway KeySrv
user@host# set group-vpn-external-interface ge-0/0/1.0
user@host# set group 1
user@host# set recovery-probe
```

5. Configure the IPsec policy.

```
[edit security ipsec-policy from-zone LAN to-zone WAN]
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_LAN;
        family inet {
            address 172.16.102.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_KeySrv;
        family inet {
            address 10.18.102.1/24;
        }
    }
}
[edit]
user@host# show routing-options
static {
    route 10.18.101.0/24 next-hop 10.18.102.254;
    route 10.18.103.0/24 next-hop 10.18.102.254;
    route 10.18.104.0/24 next-hop 10.18.102.254;
    route 172.16.101.0/24 next-hop 10.18.102.254;
    route 172.16.102.0/24 next-hop 10.18.102.254;
    route 172.16.103.0/24 next-hop 10.18.102.254;
    route 172.16.104.0/24 next-hop 10.18.102.254;
    route 10.10.100.0/24 next-hop 10.18.102.254;
}
[edit]
user@host# show security
address-book {
    global {
        address 172.16.0.0/12 172.16.0.0/12;
    }
```

```
}
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy KeySrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            gateway KeySrv {
                ike-policy KeySrv;
                server-address 10.10.100.1;
                local-address 10.18.102.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway KeySrv;
                group-vpn-external-interface ge-0/0/1.0;
                group 1;
                recovery-probe;
            }
        }
    }
}
policies {
    from-zone LAN to-zone WAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    from-zone WAN to-zone LAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                reject;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone LAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone WAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Group Member GM-0003 (MX Series Device)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet address 10.18.103.1/24
set interfaces xe-0/0/2 unit 0 family inet address 172.16.103.1/24
set interfaces ms-0/2/0 unit 0 family inet
set routing-options static route 10.18.101.0/24 next-hop 10.18.103.254
set routing-options static route 10.18.102.0/24 next-hop 10.18.103.254
set routing-options static route 10.18.104.0/24 next-hop 10.18.103.254
set routing-options static route 172.16.101.0/24 next-hop 10.18.103.254
set routing-options static route 172.16.102.0/24 next-hop 10.18.103.254
set routing-options static route 172.16.103.0/24 next-hop 10.18.103.254
set routing-options static route 172.16.104.0/24 next-hop 10.18.103.254
set routing-options static route 10.10.100.0/24 next-hop 10.18.103.254
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy KeySrv mode main
set security group-vpn member ike policy KeySrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy KeySrv pre-shared-key ascii-text "$ABC123"
set security group-vpn member ike gateway KeySrv ike-policy KeySrv
set security group-vpn member ike gateway KeySrv server-address 10.10.100.1
set security group-vpn member ike gateway KeySrv local-address 10.18.103.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway KeySrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 match-direction output
set security group-vpn member ipsec vpn GROUP_ID-0001 tunnel-mtu 1400
set security group-vpn member ipsec vpn GROUP_ID-0001 df-bit clear
set services service-set GROUP_ID-0001 interface-service service-interface ms-0/2/0.0
set services service-set GROUP_ID-0001 ipsec-group-vpn GROUP_ID-0001
set firewall family inet service-filter GroupVPN-KS term inbound-ks from destination-address
10.10.100.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.10.100.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.10.100.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from source-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from destination-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 then service
```

Step-by-Step Procedure

To configure the Group VPNv2 member:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet address 10.18.103.1/24
user@host# set xe-0/0/2 unit 0 family inet address 172.16.103.1/24
user@host# set ms-0/2/0 unit 0 family inet
```

2. Configure routing.

```
[edit routing-options]
user@host# set static route 10.18.101.0/24 next-hop 10.18.103.254
user@host# set static route 10.18.102.0/24 next-hop 10.18.103.254
user@host# set static route 10.18.104.0/24 next-hop 10.18.103.254
user@host# set static route 172.16.101.0/24 next-hop 10.18.103.254
user@host# set static route 172.16.102.0/24 next-hop 10.18.103.254
user@host# set static route 172.16.103.0/24 next-hop 10.18.103.254
user@host# set static route 172.16.104.0/24 next-hop 10.18.103.254
user@host# set static route 10.10.100.0/24 next-hop 10.18.103.254
```

3. Configure IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256 ]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy KeySrv ]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn member ike gateway KeySrv]
user@host# set ike-policy KeySrv
```

```
user@host# set server-address 10.10.100.1
user@host# set local-address 10.18.103.1
```

4. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway KeySrv
user@host# set group 1
user@host# set match-direction output
user@host# set tunnel-mtu 1400
user@host# set df-bit clear
```

5. Configure the service filter.

```
[edit firewall family inet service-filter GroupVPN-KS]
user@host# set term inbound-ks from destination-address 10.10.100.1/32
user@host# set term inbound-ks from source-address 10.10.100.1/32
user@host# set term inbound-ks then skip
user@host# set term outbound-ks from destination-address 10.10.100.1/32
user@host# set term outbound-ks then skip
user@host# set term GROUP_ID-0001 from source-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 from destination-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 then service
```

6. Configure the service set.

```
[edit services service-set GROUP_ID-0001]
user@host# set interface-service service-interface ms-0/2/0.0
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security`, `show services`, and `show firewall` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
```

```
xe-0/0/1 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
                output {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
            }
            address 10.18.103.1/24;
        }
    }
}
xe-0/0/2 {
    unit 0 {
        family inet {
            address 172.16.103.1/24;
        }
    }
}
ms-0/2/0 {
    unit 0 {
        family inet;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.18.101.0/24 next-hop 10.18.103.254;
    route 10.18.102.0/24 next-hop 10.18.103.254;
    route 10.18.104.0/24 next-hop 10.18.103.254;
    route 172.16.101.0/24 next-hop 10.18.103.254;
    route 172.16.102.0/24 next-hop 10.18.103.254;
    route 172.16.103.0/24 next-hop 10.18.103.254;
    route 172.16.104.0/24 next-hop 10.18.103.254;
}
[edit]
user@host# show security
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy KeySrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            gateway KeySrv {
                ike-policy KeySrv;
                local-address 10.18.103.1;
                server-address 10.10.101.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway KeySrv
                group 1;
                match-direction output;
                tunnel-mtu 1400;
                df-bit clear;
            }
        }
    }
}
[edit]
user@host# show services
service-set GROUP_ID-0001 {
    interface-service {
        service-interface ms-0/2/0.0;
    }
    ipsec-group-vpn GROUP_ID-0001;
}
[edit]
user@host# show firewall
family inet {
    service-filter GroupVPN-KS {
        term inbound-ks {
            from {
                destination-address {
                    10.10.100.1/32;
                }
                source-address {
                    10.10.100.1/32;
                }
            }
            then skip;
        }
        term outbound-ks {
            from {
                destination-address {
                    10.10.100.1/32;
                }
            }
            then skip;
        }
        term GROUP_ID-0001 {
            from {
                source-address {
                    172.16.0.0/12;
                }
                destination-address {
                    172.16.0.0/12;
                }
            }
            then service;
        }
    }
}
```

Configuring Group Member GM-0004 (MX Series Device)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
```

```
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet address 10.18.104.1/24
set interfaces xe-0/0/2 unit 0 family inet address 172.16.104.1/24
set interfaces ms-0/2/0 unit 0 family inet
set routing-options static route 10.18.101.0/24 next-hop 10.18.104.254
set routing-options static route 10.18.102.0/24 next-hop 10.18.104.254
set routing-options static route 10.18.103.0/24 next-hop 10.18.104.254
set routing-options static route 172.16.101.0/24 next-hop 10.18.104.254
set routing-options static route 172.16.102.0/24 next-hop 10.18.104.254
set routing-options static route 172.16.103.0/24 next-hop 10.18.104.254
set routing-options static route 172.16.104.0/24 next-hop 10.18.104.254
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy SubSrv mode main
set security group-vpn member ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy SubSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn member ike gateway SubSrv ike-policy SubSrv
set security group-vpn member ike gateway SubSrv server-address 10.17.101.1
set security group-vpn member ike gateway SubSrv server-address 10.17.102.1
set security group-vpn member ike gateway SubSrv server-address 10.17.103.1
set security group-vpn member ike gateway SubSrv server-address 10.17.104.1
set security group-vpn member ike gateway SubSrv local-address 10.18.104.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway SubSrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 match-direction output
set security group-vpn member ipsec vpn GROUP_ID-0001 tunnel-mtu 1400
set security group-vpn member ipsec vpn GROUP_ID-0001 df-bit clear
set services service-set GROUP_ID-0001 interface-service service-interface ms-0/2/0.0
set services service-set GROUP_ID-0001 ipsec-group-vpn GROUP_ID-0001
set firewall family inet service-filter GroupVPN-KS term inbound-ks from destination-address
10.10.100.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.10.100.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.101.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.102.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
```

```
10.17.103.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.104.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from source-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from destination-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 then service
```

Step-by-Step Procedure

To configure the Group VPNv2 member:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet address 10.18.104.1/24
user@host# set xe-0/0/2 unit 0 family inet address 172.16.104.1/24
user@host# set ms-0/2/0 unit 0 family inet
```

2. Configure routing.

```
[edit routing-options]
user@host# set static route 10.18.101.0/24 next-hop 10.18.104.254
user@host# set static route 10.18.102.0/24 next-hop 10.18.104.254
user@host# set static route 10.18.103.0/24 next-hop 10.18.104.254
user@host# set static route 172.16.101.0/24 next-hop 10.18.104.254
user@host# set static route 172.16.102.0/24 next-hop 10.18.104.254
user@host# set static route 172.16.103.0/24 next-hop 10.18.104.254
user@host# set static route 172.16.104.0/24 next-hop 10.18.104.254
```

3. Configure IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256 ]
user@host# set authentication-method pre-shared-keys
```

```
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy KeySrv ]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn member ike gateway KeySrv]
user@host# set ike-policy KeySrv
user@host# set server-address 10.10.100.1
user@host# set local-address 10.18.104.1
```

4. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway KeySrv
user@host# set group 1
user@host# set match-direction output
user@host# set tunnel-mtu 1400
user@host# set df-bit clear
```

5. Configure the service filter.

```
[edit firewall family inet service-filter GroupVPN-KS]
user@host# set term inbound-ks from destination-address 10.10.101.1/32
user@host# set term inbound-ks from source-address 10.10.101.1/32
user@host# set term inbound-ks then skip
user@host# set term outbound-ks from destination-address 10.17.101.1/32
user@host# set term outbound-ks from destination-address 10.17.102.1/32
user@host# set term outbound-ks from destination-address 10.17.103.1/32
user@host# set term outbound-ks from destination-address 10.17.104.1/32
user@host# set term outbound-ks then skip
user@host# set term GROUP_ID-0001 from source-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 from destination-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 then service
```

6. Configure the service set.

```
[edit services service-set GROUP_ID-0001]
user@host# set interface-service service-interface ms-0/2/0.0
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show routing-options`, `show security`, `show services`, and `show firewall` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
xe-0/0/1 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
                output {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
            }
            address 10.18.104.1/24;
        }
    }
}
xe-0/0/2 {
    unit 0 {
        family inet {
            address 172.16.104.1/24;
        }
    }
}
ms-0/2/0 {
    unit 0 {
        family inet;
    }
```

```
}
[edit]
user@host# show routing-options
static {
    route 10.18.101.0/24 next-hop 10.18.104.254;
    route 10.18.102.0/24 next-hop 10.18.104.254;
    route 10.18.103.0/24 next-hop 10.18.104.254;
    route 172.16.101.0/24 next-hop 10.18.104.254;
    route 172.16.102.0/24 next-hop 10.18.104.254;
    route 172.16.103.0/24 next-hop 10.18.104.254;
    route 172.16.104.0/24 next-hop 10.18.104.254;
}
[edit]
user@host# show security
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy KeySrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            gateway KeySrv {
                ike-policy KeySrv;
                local-address 10.18.104.1;
                server-address 10.17.101.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway KeySrv
                group 1;
                match-direction output;
                tunnel-mtu 1400;
                df-bit clear;
            }
        }
    }
}
[edit]
user@host# show services
service-set GROUP_ID-0001 {
    interface-service {
        service-interface ms-0/2/0.0;
    }
    ipsec-group-vpn GROUP_ID-0001;
}
[edit]
user@host# show firewall
family inet {
    service-filter GroupVPN-KS {
        term inbound-ks {
            from {
                destination-address {
                    10.10.100.1/32;
                }
                source-address {
                    10.10.100.1/32;
                }
            }
            then skip;
        }
        term outbound-ks {
            from {
                destination-address {
                    10.17.101.1/32;
                    10.17.102.1/32;
                    10.17.103.1/32;
                    10.17.104.1/32;
                }
            }
            then skip;
        }
        term GROUP_ID-0001 {
            from {
                source-address {
                    172.16.0.0/12;
                }
                destination-address {
                    172.16.0.0/12;
                }
            }
            then service;
        }
    }
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Group Member Registration

Purpose

Verify that group members are registered on the server.

Action

From operational mode, enter the `show security group-vpn server registered-members` and `show security group-vpn server registered-members detail` commands on the server.

```
user@host> show security group-vpn server registered-members
Group: GROUP_ID-0001, Group Id: 1
  Total number of registered members: 2
  Member Gateway                   Member IP       Last Update              Vsys
  GM-0001                          10.18.101.1     Thu Nov 19 2015 16:31:09 root
  GM-0003                          10.18.103.1     Thu Nov 19 2015 16:29:47 root
```

```
user@host> show security group-vpn server registered-members detail
GGroup: GROUP_ID-0001, Group Id: 1
  Total number of registered members: 2

  Member gateway: GM-0001, Member IP: 10.18.101.1, Vsys: root
  Last Update: Thu Nov 19 2015 16:31:09
  Stats:
      Pull Succeeded                : 2
      Pull Failed                   : 0
      Push Sent                     : 0
      Push Acknowledged             : 0
      Push Unacknowledged           : 0

  Member gateway: GM-0003, Member IP: 10.18.103.1, Vsys: root
  Last Update: Thu Nov 19 2015 16:29:47
  Stats:
      Pull Succeeded                : 1
      Pull Failed                   : 0
      Push Sent                     : 0
      Push Acknowledged             : 0
      Push Unacknowledged           : 0
```

Verifying That Group Keys Are Distributed

Purpose

Verify that group keys are distributed to members.

Action

From operational mode, enter the `show security group-vpn server statistics` command on the group server.

```
user@host> show security group-vpn server statistics
Group: GROUP_ID-0001, Group Id: 1
  Stats:
      Pull Succeeded                : 4
      Pull Failed                   : 0
      Pull Exceed Member Threshold  : 0
      Push Sent                     : 0
      Push Acknowledged             : 0
      Push Unacknowledged           : 0
```

Verifying Group VPN SAs on the Group Server

Purpose

Verify Group VPN SAs on the group server.

Action

From operational mode, enter the `show security group-vpn server kek security-associations` and `show security group-vpn server kek security-associations detail` commands on the group server.

```
user@host> show security group-vpn server kek security-associations
Index   Life:sec  Initiator cookie  Responder cookie  GroupId
738879  1206      a471513492db1e13  24045792a4b3dd64  1
```

```
user@host> show security group-vpn server kek security-associations detail
Index 738879, Group Name: GROUP_ID-0001, Group Id: 1
Initiator cookie: a471513492db1e13, Responder cookie: 24045792a4b3dd64
Authentication method: RSA
Lifetime: Expires in 1204 seconds, Activated
Rekey in 694 seconds
  Algorithms:
   Sig-hash              : sha256
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  Server Member Communication: Unicast
  Retransmission Period: 10, Number of Retransmissions: 2
  Group Key Push sequence number: 0

PUSH negotiations in progress: 0
```

Verifying Group VPN SAs on Group Members

Purpose

Verify Group VPN SAs on the group members.

Action

From operational mode, enter the `show security group-vpn member kek security-associations` and `show security group-vpn member kek security-associations detail` commands on the SRX Series Firewall or vSRX Virtual Firewall group member.

```
user@host> show security group-vpn member kek security-associations
Index   Server Address  Life:sec  Initiator cookie  Responder cookie  GroupId
5455810 10.10.100.1     1093      a471513492db1e13  24045792a4b3dd64  1
```

```
user@host> show security group-vpn member kek security-associations detail
  Index 5455810, Group Id: 1
  Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.101.1, GDOI Server: 10.10.100.1
  Initiator cookie: a471513492db1e13, Responder cookie: 24045792a4b3dd64
  Lifetime: Expires in 1090 seconds
  Group Key Push Sequence number: 0

  Algorithms:
   Sig-hash              : hmac-sha256-128
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  Stats:
      Push received            :   0
      Delete received          :   0
```

From operational mode, enter the `show security group-vpn member kek security-associations` and `show security group-vpn member kek security-associations detail` commands on the MX Series group member.

```
user@host> show security group-vpn member kek security-associations
Index   Server Address  Life:sec  Initiator cookie  Responder cookie  GroupId
488598  10.10.100.1     963       a471513492db1e13  24045792a4b3dd64  1
```

```
user@host> show security group-vpn member kek security-associations detail
  Index 488598, Group Id: 1
  Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.103.1, GDOI Server: 10.10.100.1
  Initiator cookie: a471513492db1e13, Responder cookie: 24045792a4b3dd64
  Lifetime: Expires in 961 seconds
  Group Key Push Sequence number: 0

  Algorithms:
   Sig-hash             : hmac-sha256-128
   Encryption           : aes256-cbc
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
  Stats:
      Push received          :   0
      Delete received        :   0
```

Verifying IPsec SAs on the Group Server

Purpose

Verify IPsec SAs on the group server.

Action

From operational mode, enter the `show security group-vpn server ipsec security-associations` and `show security group-vpn server ipsec security-associations detail` commands on the group server.

```
user@host> show security group-vpn server ipsec security-associations
Group: GROUP_ID-0001, Group Id: 1
  Total IPsec SAs: 1
  IPsec SA          Algorithm        SPI              Lifetime
  GROUP_ID-0001     ESP:aes-256/sha256 1c548e4e       1156
```

```
user@host> show security group-vpn server ipsec security-associations detail
Group: GROUP_ID-0001, Group Id: 1
Total IPsec SAs: 1
  IPsec SA: GROUP_ID-0001
    Protocol: ESP, Authentication: sha256, Encryption: aes-256
    Anti-replay: D3P enabled
    SPI: 1c548e4e
    Lifetime: Expires in 1152 seconds, Activated
    Rekey in 642 seconds
    Policy Name: 1
      Source: 172.16.0.0/12
      Destination: 172.16.0.0/12
      Source Port: 0
      Destination Port: 0
      Protocol: 0
```

Verifying IPsec SAs on the Group Members

Purpose

Verify IPsec SAs on the group members.

Action

From operational mode, enter the `show security group-vpn member ipsec security-associations` and `show security group-vpn member ipsec security-associations detail` commands on the SRX Series Firewall or vSRX Virtual Firewall group member.

```
user@host> show security group-vpn member ipsec security-associations
  Total active tunnels: 1
  ID    Server           Port  Algorithm       SPI      Life:sec/kb  GId lsys
  <>49152 10.10.100.1    848   ESP:aes-256/sha256-128 1c548e4e 1073/ unlim 1 root
```

```
user@host> show security group-vpn member ipsec security-associations detail
  Virtual-system: root Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.101.1, GDOI Server: 10.10.100.1
  Group Id: 1
  Routing Instance: default
  Recovery Probe: Enabled
  DF-bit: clear
  Stats:
      Pull Succeeded             :   4
      Pull Failed                :   3
      Pull Timeout               :   3
      Pull Aborted               :   0
      Push Succeeded             :   6
      Push Failed                :   0
      Server Failover            :   0
      Delete Received            :   0
      Exceed Maximum Keys(4)     :   0
      Exceed Maximum Policies(10):   0
      Unsupported Algo           :   0
  Flags:
      Rekey Needed:   no

    List of policies received from server:
    Tunnel-id: 49152
      Source IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)
      Destination IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)

      Direction: bi-directional, SPI: 1c548e4e
      Protocol: ESP, Authentication: sha256-128, Encryption: aes-256
      Hard lifetime: Expires in 1070 seconds, Activated
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expires in 931 seconds
      Mode: Tunnel, Type: Group VPN, State: installed
      Anti-replay service: D3P enabled
```

From operational mode, enter the `show security group-vpn member ipsec security-associations` and `show security group-vpn member ipsec security-associations detail` commands on the MX Series group member.

```
user@host> show security group-vpn member ipsec security-associations
  Total active tunnels: 1
  ID    Server           Port  Algorithm       SPI      Life:sec/kb  GId lsys
  <>10001 10.10.100.1    848   ESP:aes-256/sha256-128 1c548e4e 947/ unlim 1 root
```

```
user@host> show security group-vpn member ipsec security-associations detail
  Virtual-system: root Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.103.1, GDOI Server: 10.10.100.1
  Group Id: 1
  Rule Match Direction: output,  Tunnel-MTU: 1400
  Routing Instance: default
  DF-bit: clear
  Stats:
      Pull Succeeded            :   2
      Pull Failed               :   0
      Pull Timeout              :   1
      Pull Aborted              :   0
      Push Succeeded            :   2
      Push Failed               :   0
      Server Failover           :   0
      Delete Received           :   0
      Exceed Maximum Keys(4)    :   0
      Exceed Maximum Policies(1):   0
      Unsupported Algo          :   0
  Flags:
      Rekey Needed:   no

    List of policies received from server:
    Tunnel-id: 10001
      Source IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)
      Destination IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)

      Direction: bi-directional, SPI: 1c548e4e
      Protocol: ESP, Authentication: sha256-128, Encryption: aes-256
      Hard lifetime: Expires in 945 seconds, Activated
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expires in 840 seconds
      Mode: Tunnel, Type: Group VPN, State: installed
      Anti-replay service: D3P enabled
```

Verifying Group Policies (SRX Series Firewall or vSRX Virtual Firewall Group Members Only)

Purpose

Verify group policies on SRX Series Firewall or vSRX Virtual Firewall group members.

Action

From operational mode, enter the `show security group-vpn member policy` command on the group member.

```
user@host> show security group-vpn member policy
Group VPN Name: GROUP_ID-0001, Group Id: 1
From-zone: LAN, To-zone: WAN
 Tunnel-id: 49152, Policy type: Secure
  Source      : IP <172.16.0.0 - 172.31.255.255>, Port <0 - 65535>, Protocol <0>
  Destination : IP <172.16.0.0 - 172.31.255.255>, Port <0 - 65535>, Protocol <0>

 Tunnel-id: 63488, Policy type: Fail-close
  Source      : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <0>
  Destination : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <0>
```

SEE ALSO

Configuring Group VPNs in Group VPNv2 on Routing Device

Example: Configuring Group VPNv2 Server-Member Communication for Unicast Rekey Messages

IN THIS SECTION

This example shows how to enable the server to send unicast rekey messages to group members to ensure that valid keys are available for encrypting traffic between group members. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

Requirements

Before you begin:

- Configure the group server and members for IKE Phase 1 negotiation.
- Configure the group server and members for IPsec SA.
- Configure the group `g1` on the group server.

Overview

In this example, you specify the following server-member communication parameters for group `g1`:

- The server sends unicast rekey messages to group members.
- aes-128-cbc is used to encrypt traffic between the server and members.
- sha-256 is used for member authentication.

Default values are used for KEK lifetime and retransmissions.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure server-member communication:

1. Set the communications type.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set communications-type unicast
```

2. Set the encryption algorithm.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set encryption-algorithm aes-128-cbc
```

3. Set the member authentication.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set sig-hash-algorithm sha-256
```

Verification

To verify the configuration is working properly, enter the `show security group-vpn server group g1 server-member-communication` command.

SEE ALSO

Example: Configuring Group VPNv1 Server and Members | 717
Example: Configuring Group VPNv1 with Server-Member Colocation | 744

RELATED DOCUMENTATION

Monitoring VPN Traffic | 1351
VPN Session Affinity | 1374

Group VPNv2 Server Clusters

IN THIS SECTION

Group VPNv2 server cluster provides group controller/key server (GCKS) redundancy, so there is no single point of failure for the entire group VPN network.

Understanding Group VPNv2 Server Clusters

IN THIS SECTION

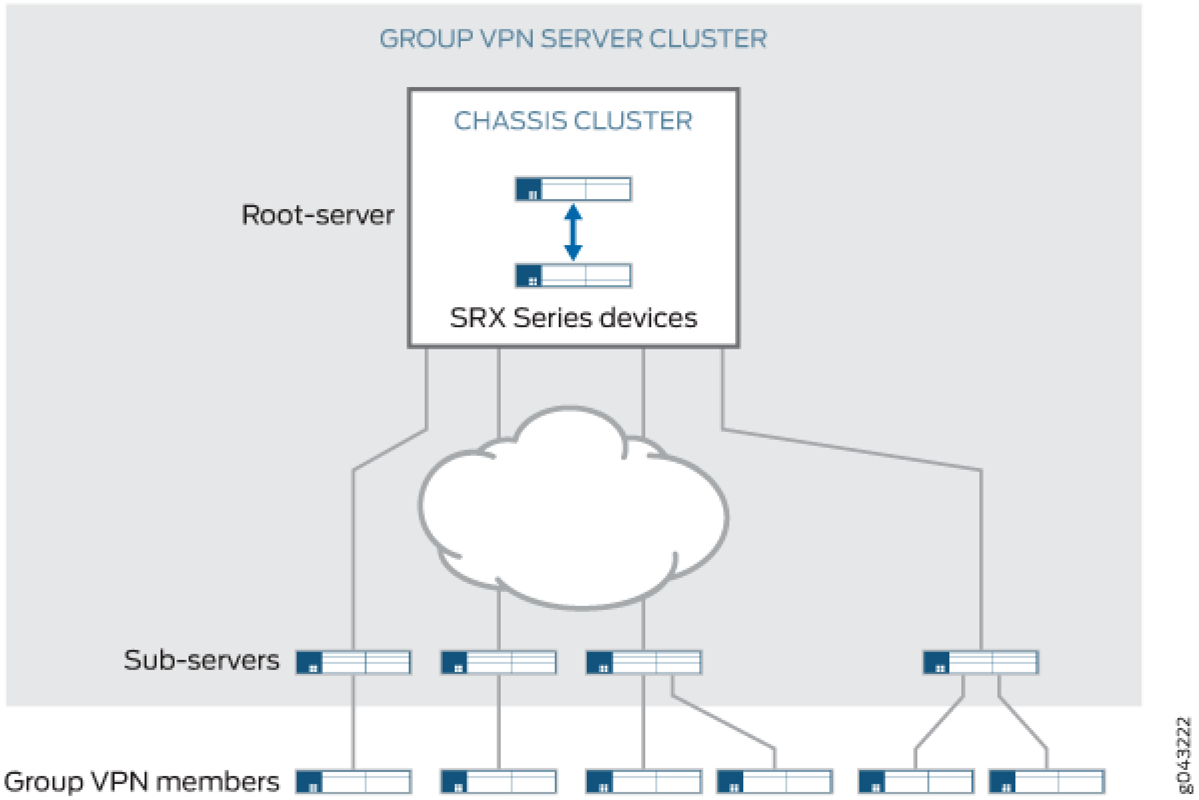
In the Group Domain of Interpretation (GDOI) protocol, the group controller/key server (GCKS) manages Group VPN security associations (SAs), and generates encryption keys and distributes them to group members. Group members encrypt traffic based on the group SAs and keys provided by the GCKS. If the GCKS fails, group members cannot register or obtain keys. A *Group VPNv2 server cluster* provides GCKS redundancy so there is no single point of failure for the entire group VPN network. Group VPNv2 server clusters can also provide load balancing, scaling, and link redundancy.

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. All servers in a Group VPNv2 server cluster must be supported on SRX Series Firewalls or vSRX Virtual Firewall instances. Group VPNv2 server clusters are a Juniper Networks proprietary solution and have no interoperability with other vendor's GCKS.

Root-Server and Sub-Servers

A Group VPNv2 server cluster consists of one root-server with up to four connected sub-servers. All servers in the cluster share the same SA and encryption keys that are distributed to Group VPNv2 members. Servers in the cluster can be located at different sites, as shown in Figure 55 on page 820.

Figure 55: Group VPNv2 Server Cluster



Messages between servers in the cluster are encrypted and authenticated by IKE SAs. The root-server is responsible for generating and distributing encryption keys to sub-servers; because of this responsibility, we recommend that the root-server be configured as a chassis cluster. Sub-servers are single devices and cannot be chassis clusters. Sub-servers must be able to connect to the root-server, although direct links between sub-servers are not necessary.

If a sub-server loses its connection to the root-server, no further connection to the sub-server from group members are allowed and SAs are deleted. Therefore, we recommend that you use a different link to connect each sub-server to the root-server.

Group VPNv2 server clusters are configured with the `server-cluster` statements at the [`edit security group-vpn server` *`group-name`*] hierarchy level. The following values must be configured for each server in a cluster:

- The server role—Specify either `root-server` or `sub-server`. A given server can be part of multiple Group VPNv2 server clusters, but it must have the same server role in all clusters. A server cannot be configured with the root-server role in one group and the sub-server role in another group.

  You must ensure that there is only one root-server at any time for a Group VPNv2 server cluster.

- IKE gateway—Specify the name of an IKE gateway configured at the [edit security group-vpn server ike] hierarchy level. For a root-server, the IKE gateway must be a sub-server in the cluster; up to four sub-servers can be specified. For sub-servers, the IKE gateway must be the root-server.

  The root-server and sub-servers must be configured with dead-peer-detection always-send and cannot be configured for a dynamic (unspecified) IP address. Group members are not configured with dead peer detection.

The Group VPNv2 configuration must be the same on each sub-server in a given group.

Each sub-server in the Group VPNv2 server cluster operates as a normal GCKS for registering and deleting members. Upon successful member registration, the registering server is responsible for sending updates to the member. For a given group, you can configure the maximum number of Group VPNv2 members that can be accepted by each sub-server; this number must be the same on all sub-servers in the cluster. A sub-server stops responding to registration requests by new members when it reaches the configured maximum number of Group VPNv2 members. See "Load Balancing" on page 822.

Group Member Registration with Server Clusters

Group members can register with any server in the Group VPNv2 server cluster for a given group, however we recommend that members only connect to sub-servers and not the root-server. Up to four server addresses can be configured on each group member. The server addresses configured on group members can be different. In the example shown below, group member A is configured for sub-servers 1 through 4, while member B is configured for sub-servers 4 and 3:

	Group member A:	Group member B:
Server addresses:	Sub-server 1 Sub-server 2 Sub-server 3 Sub-server 4	Sub-server 4 Sub-server 3

The order that the server addresses is configured on a member is important. A group member attempts to register with the first configured server. If registration with a configured server is not successful, the group member tries to register with the next configured server.

Each server in a Group VPNv2 server cluster operates as a normal GCKS for registering and deleting members. Upon successful registration, the registering server is responsible for sending updates to the member via groupkey-push exchanges. For a given group, you can configure the maximum number of group members that can be accepted by each server, however this number must be the same on all servers in the cluster for a given group. Upon reaching the configured maximum number of group members, a server stops responding to registration requests by new members. See "Load Balancing" on page 822 for additional information.

Dead Peer Detection

To verify the availability of peer servers in a Group VPNv2 server cluster, each server in the cluster must be configured to send dead peer detection (DPD) requests regardless of whether there is outgoing IPsec traffic to the peer. This is configured with the `dead-peer-detection always-send` statement at the [`edit security group-vpn server ike gateway` *`gateway-name`*] hierarchy level.

An active server in a Group VPNv2 server cluster sends DPD probes to the IKE gateway(s) configured in the server cluster. DPD should not be configured for a group because multiple groups can share the same peer server IKE gateway configuration. When DPD detects that a server is down, the IKE SA with that server is deleted. All groups mark the server as inactive and DPD to the server is stopped.

DPD should not be configured for the IKE gateway on group members.

When DPD marks the root-server as inactive, the sub-servers stop responding to new group member requests however existing SAs for current group members remain active. An inactive sub-server does not send deletes to group members because the SAs could be still valid and group members can continue using existing SAs.

If an IKE SA expires while a peer server is still active, DPD triggers IKE SA negotiation. Because both root-servers and sub-servers can trigger IKE SAs through DPD, simultaneous negotiation might result in multiple IKE SAs. No impact on server-cluster functionality is expected in this case.

Load Balancing

Load balancing in the Group VPNv2 server cluster can be achieved by configuring the right `member-threshold` value for the group. When the number of members registered on a server exceeds the `member-threshold` value, subsequent member registration on that server is rejected. The member registration fails over to the next server configured on the group member until it reaches a server whose `member-threshold` is not yet reached.

There are two restrictions on configuring the `member-threshold`:

- For a given group, the same `member-threshold` value must be configured on the root-server and all sub-servers in a group server cluster. If the total number of members in the group exceeds the configured `member-threshold` value, then a `groupkey-pull` registration initiated by a new member is rejected (the server does not send a response).

- A server can support members in multiple groups. Each server has a maximum number of group members that it can support. If a server reaches the maximum number of members it can support, then a `groupkey-pull` registration initiated by a new member is rejected even if the `member-threshold` value of a specific group has not been reached.

There is no member synchronization among servers in the cluster. The root-server does not have information about the number of registered members on sub-servers. Each sub-server can only show its own registered members.

SEE ALSO

Group VPNv2 Overview | 758

Understanding Group VPNv2 Server Cluster Limitations

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. Note the following caveats when configuring Group VPNv2 server clusters:

- Certificate authentication is not supported for server authentication; only preshared keys can be configured.
- There is no configuration synchronization between servers in the Group VPNv2 server cluster.
- When enabling a Group VPNv2 server cluster, configuration must be done on the root-server first and then on the sub-servers. Until the configuration is manually synchronized among the servers, traffic loss can be expected during the configuration change.
- In certain corner cases, the SAs on Group VPNv2 members can be out of sync. Group VPN members can synchronize SAs by getting a new key through a `groupkey-pull` exchange. You can manually clear SAs on a Group VPNv2 member with the `clear security group-vpn member ipsec security-associations` or `clear security group-vpn member group` commands to help speed recovery.
- The Group VPNv2 server cluster does not support ISSU.
- If the last `groupkey-pull` message is lost during a Group VPNv2 member's registration, a server might consider the member to be a registered member even though the member might fail over to the next server in the server cluster. In this case, the same member might appear to be registered on multiple servers. If the total member-threshold on all servers equals the total number of deployed members, subsequent group members might fail to register.

Note the following caveats for chassis cluster operations on the root-server:

- No statistics are preserved.

- No negotiation data or state is saved. If a root-server chassis cluster failover occurs during a `groupkey-pull` or `groupkey-push` negotiation, the negotiation is not restarted after the failover.

- If both chassis cluster nodes of a root-server go down during a rekey of an encryption key, some Group VPNv2 members might receive the new key while other members do not. Traffic might be impacted. Manually clearing SAs on a Group VPNv2 member with the `clear security group-vpn member ipsec security-associations` or `clear security group-vpn member group` commands might help speed up recovery when the root-server becomes reachable.

- In a large-scale environment, RG0 failover on the root-server might take time. If the DPD interval and threshold on a sub-server are configured with small values, it can result in the sub-server marking the root-server as inactive during an RG0 failover. Traffic might be impacted. We recommend that you configure the IKE gateway for the sub-server with a DPD `interval` * `threshold` value larger than 150 seconds.

Understanding Group VPNv2 Server Cluster Messages

IN THIS SECTION

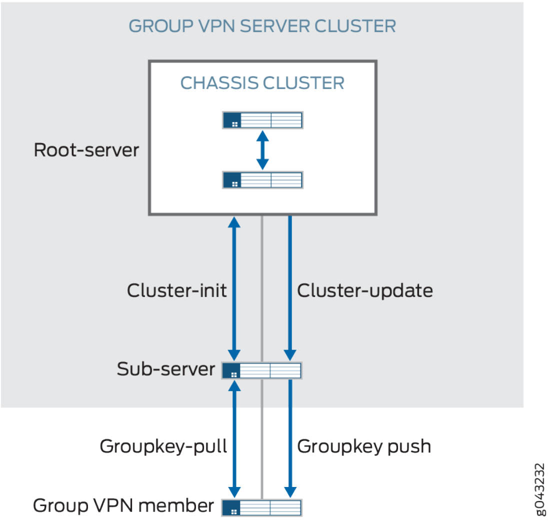
Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. All messages between servers in a Group VPNv2 server cluster are encrypted and authenticated by an IKE security association (SA). Each sub-server initiates an IKE SA with the root-server; this IKE SA must be established before messages can be exchanged between the servers.

This section describes the messages exchanged between the root-server and sub-servers.

Cluster Exchanges

Figure 56 on page 825 shows the basic messages exchanged between the Group VPNv2 server cluster and Group VPNv2 members.

Figure 56: Group VPNv2 Server Cluster Messages



Cluster-Init Exchanges

A sub-server launches a cluster initialization (`cluster-init`) exchange with the root-server to obtain SA and encryption key information. The root-server responds by sending current SA information to the sub-server through the `cluster-init` exchange.

Sub-servers can then respond to registration requests from Group VPNv2 members through a `groupkey-pull` exchange. The `groupkey-pull` exchange allows a Group VPNv2 member to request SAs and keys shared by the group from a sub-server.

Sub-servers start a `cluster-init` exchange with the root-server when:

- The root-server is considered inactive. This is the initial assumed state of the root-server. If there is no IKE SA between the root-server and the sub-server, the sub-server initiates an IKE SA with the root-server. After a successful `cluster-init` exchange, the sub-server obtains information on SAs and marks the root-server as active.

- The soft lifetime of the SA has expired.

- A `cluster-update` message is received to delete all SAs.

- There are group configuration changes.

If the `cluster-init` exchange fails, the sub-server retries the exchange with the root-server every 5 seconds.

Cluster-Update Messages

The groupkey-push exchange is a single rekey message that allows a group controller/key server (GCKS) to send group SAs and keys to members before existing group SAs expire and to update group membership. Rekey messages are unsolicited messages sent from the GCKS to members

Upon generating new encryption keys for an SA, the root-server sends SA updates to all active sub-servers through a cluster-update message. After receiving a cluster-update from the root-server, the sub-server installs the new SA and sends the new SA information through a groupkey-push to its registered group members.

A cluster-update message sent from the root-server requires an acknowledgement from the sub-server. If there is no acknowledgement received from a sub-server, the root-server retransmits the cluster-update at the configured retransmission period (the default is 10 seconds). The root-server does not retransmit if dead peer detection (DPD) indicates that the sub-server is unavailable. If a sub-server fails to update SA information after receiving a cluster-update, it does not send an acknowledgement and the root-server retransmits the cluster-update message.

If the soft lifetime of an SA expires before a new SA is received from the root-server, the sub-server sends a cluster-init message to the root-server to get all SAs and does not send a groupkey-push message to its members until it has a new update. If the hard lifetime of an SA expires on the sub-server before it receives a new SA, the sub-server marks the root-server inactive, deletes all registered group members, and continues to send cluster-init messages to the root-server.

A cluster-update message can be sent to delete an SA or a group member; this can be the result of a clear command or a configuration change. If a sub-server receives a cluster-update message to delete an SA, it sends a groupkey-push delete message to its group members and deletes the corresponding SA. If all SAs for a group are deleted, the sub-server initiates a cluster-init exchange with the root-server. If all registered members are deleted, the sub-server deletes all locally registered members.

Understanding Configuration Changes with Group VPNv2 Server Clusters

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances. Group VPNv2 server clusters behave differently from standalone Group VPNv2 servers when there are configuration changes that result in new encryption keys and changes to security associations (SAs). The root-server sends SA updates or deletions to sub-servers through cluster-update messages. The sub-servers then send groupkey-push messages to members. Sub-servers cannot send delete messages to group members without first receiving delete messages from the root-server.

All configuration changes must be made on the root-server first and then on sub-servers to ensure that group members receive updates or deletions as expected. Until configuration is synchronized between the servers in the Group VPNv2 server cluster, traffic loss can be expected.

Table 81 on page 827 describes the effects of various configuration changes on Group VPNv2 servers.

Table 81: Effects of Configuration Changes on Group VPNv2 Servers

Configuration Change	Standalone Group VPNv2 Server Action	Group VPNv2 Server Cluster Action	
		Root-server	Sub-server
Change IKE proposal, policy, or gateway	Delete the IKE SA for the affected gateway. For IKE proposal, policy, or gateway deletions, delete the registered members for the affected gateway.		
Change IPsec proposal	Changes take effect after the traffic encryption key (TEK) rekey.		
Group changes:			
Delete group name	Send “delete all” to group members. Delete all IKE SAs in the group. Delete all keys in the group immediately. Delete all registered members in the group.	Send “delete all” to sub-servers. Delete all keys in the group immediately. Mark all peers inactive. Delete sub-server IKE SAs. Delete all member IKE SAs.	Delete all member IKE SAs. Delete all keys in the group immediately. Delete all registered members in the group. Mark peer inactive. Delete peer server IKE SAs.
Change ID	Send “delete all” to all members. Delete all IKE SAs in the group. Delete all keys in the group immediately. Delete all registered members in the group. Generate new keys according to the configuration.	Send ”delete all” to sub-servers. Delete all member IKE SAs in the group. Delete all keys in the group immediately. Mark all peers inactive. Delete all peer server IKE SAs. Generate new keys according to the configuration.	Delete all member IKE SAs in the group. Delete all keys in the group immediately. Delete all registered members in the group. Mark peer inactive. Delete peer server IKE SAs. Initiate new `cluster-init` exchange.
Add or delete IKE gateway	No changes for additions. For deletions, delete the IKE SA and registered members for the affected gateway.		

Table 81: Effects of Configuration Changes on Group VPNv2 Servers *(Continued)*

Configuration Change	Standalone Group VPNv2 Server Action	Group VPNv2 Server Cluster Action	
		Root-server	Sub-server
Add or change anti-replay time window	New value takes effect after the TEK rekey.		
Add or change no anti-replay	New value takes effect after the TEK rekey.		
Server-member communication changes:			
Add	Delete all registered members. Generate key encryption key (KEK) SA.	Generate KEK SA. Send new KEK SA to sub-server. Delete all member IKE SAs.	Delete all registered members.
Change	New value takes effect after KEK rekey.		
Delete	Send delete to delete all KEK SAs. Delete KEK SA.	Send delete to sub-servers. Delete KEK SA. Delete all member IKE SAs.	Delete KEK SA.
IPsec SA:			
Add	Generate new TEK SA. Update the new TEK SA on members.	Generate new TEK SA. Send new TEK SA to sub-servers.	No action.

Table 81: Effects of Configuration Changes on Group VPNv2 Servers *(Continued)*

Configuration Change	Standalone Group VPNv2 Server Action	Group VPNv2 Server Cluster Action	
		Root-server	Sub-server
Change	New value takes effect after TEK rekey. If the match-policy changes, the current TEK is removed immediately and delete groupkey-push is sent because members need to be explicitly notified that this configuration is removed.	If the match-policy changes, send delete to sub-servers. Delete TEK immediately.	If the match-policy changes, delete TEK immediately.
Delete	Delete TEK immediately. Send delete to delete this TEK SA.	Send delete to sub-servers. Delete TEK immediately.	Delete TEK immediately.

Table 82 on page 829 describes the effects of changing Group VPNv2 server cluster configuration.

You must ensure that there is only one root-server in a server cluster at any time.

Table 82: Effects of Group VPNv2 Server Cluster Configuration Changes

Server Cluster Configuration Change	Group VPNv2 Server Cluster	
	Root-server	Sub-server
IKE proposal, policy, or gateway (cluster peer)	For additions, there is no change. For changes or deletions, delete the IKE SA for the affected peer.	
Server cluster:		

Table 82: Effects of Group VPNv2 Server Cluster Configuration Changes *(Continued)*

Server Cluster Configuration Change	Group VPNv2 Server Cluster	
	Root-server	Sub-server
Add	None.	Send "delete all" to group members. Delete all member IKE SAs in the group. Delete all TEKs and KEKs immediately in the group. Delete all registered members in the group. Send `cluster-init` to root-server.
Change role You must ensure that there is only one root-server in a server cluster at any time.	Send "delete all" to sub-servers. Delete all member IKE SAs in the group. Delete all TEKs and KEKs immediately in the group. Mark all peers inactive. Delete all peer server IKE SAs. Send `cluster-init` to root-server.	Rekey TEK. Rekey KEK. Send new keys to sub-servers. Send new keys to members.
Add peer	None.	
Delete peer	Mark peer inactive. Clear peer IKE SA.	Mark peer inactive. Clear KEK. Clear TEK. Clear peer IKE SA.
Change retransmission period	None.	
Delete server cluster	Send "delete all" to sub-servers. Delete all TEKs and KEKs immediately in the group. Mark all peers inactive. Delete all peer server IKE SAs. Generate new TEKs and KEKs according to the configuration.	Delete all member IKE SAs in the group. Delete all TEKs and KEKs immediately in the group. Delete all registered members in the group. Mark peer inactive. Delete peer server IKE SAs. Generate new TEK and KEK according to the configuration.

Migrating a Standalone Group VPNv2 Server to a Group VPNv2 Server Cluster

Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances. This section describes how to migrate a standalone Group VPNv2 server to a Group VPNv2 server cluster.

To migrate a standalone Group VPNv2 server to a root-server:

We highly recommend that the root-server be a chassis cluster.

1. Upgrade the standalone Group VPNv2 server to a chassis cluster. See Chassis Cluster User Guide for SRX Series Devices for more information

   A reboot is required during the upgrade of a standalone SRX Series Firewall to a chassis cluster node. Traffic loss is expected.
2. On the chassis cluster, add the Group VPNv2 server cluster root-server configuration. The configured server role for the cluster must be `root-server`.

   There should be no traffic loss among existing group members during the configuration change.

To add a sub-server to the Group VPNv2 server cluster:

1. On the root-server, configure both a Group VPNv2 server IKE gateway and a server cluster IKE gateway for the sub-server. SAs and existing member traffic should not be impacted.

2. On the sub-server, configure the server cluster. Remember that the Group VPNv2 configuration must be the same on each server in the cluster, with the exception of the Group VPNv2 server IKE gateways, the server role in the cluster, and the server cluster IKE gateway configurations. On the sub-server, the configured server role in the cluster must be `sub-server`. Configure a Group VPNv2 server IKE gateway and a server cluster IKE gateway for the root-server.

To delete a sub-server from the Group VPNv2 server cluster:

1. On the root-server, delete both the Group VPNv2 server IKE gateway and the server cluster IKE gateway configurations for the sub-server. SAs and existing member traffic should not be impacted.

2. Power off the sub-server.

SEE ALSO

Group VPNv2 Overview | 758

Example: Configuring a Group VPNv2 Server Cluster and Members

IN THIS SECTION

This example shows how to configure a Group VPNv2 server cluster to provide group controller/key server (GCKS) redundancy and scaling to Group VPNv2 group members. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

Requirements

The example uses the following hardware and software components:

- Eight supported SRX Series Firewalls or vSRX Virtual Firewall instances running Junos OS Release 15.1X49-D30 or later that support Group VPNv2:
  - Two devices or instances are configured to operate as a chassis cluster. The chassis cluster operates as the root-server in the Group VPNv2 server cluster. The devices or instances must have the same software version and licenses.

    The root-server is responsible for generating and distributing encryption keys to sub-servers in the group VPN server cluster; because of this responsibility, we recommend that the root-server be a chassis cluster.
  - Four other devices or instances operate as sub-servers in the Group VPNv2 server cluster.
  - Two other devices or instances operate as Group VPNv2 group members.
- Two supported MX Series devices running Junos OS Release 15.1R2 or later that support Group VPNv2. These devices operate as Group VPNv2 group members.

A hostname, a root administrator password, and management access must be configured on each SRX Series Firewall or vSRX Virtual Firewall instance. We recommend that NTP also be configured on each device.

The configurations in this example focus on what is needed for Group VPNv2 operation, based on the topology shown in Figure 57 on page 835. Some configurations, such as interface, routing, or chassis cluster setups, are not included here. For example, Group VPNv2 operation requires a working routing topology that allows client devices to reach their intended sites throughout the network; this example does not cover the configuration of static or dynamic routing.

Overview

IN THIS SECTION

In this example, the Group VPNv2 network consists of a server cluster and four members. The server cluster consists of a root-server and four sub-servers. Two of the members are SRX Series Firewalls or vSRX Virtual Firewall instances while the other two members are MX Series devices.

The group VPN SAs must be protected by a Phase 1 SA. Therefore, the group VPN configuration must include configuring IKE Phase 1 negotiations on the root-server, the sub-servers, and the group members. IKE configurations are described as follows.

On the root-server:

- The IKE policy `SubSrv` is used to establish Phase 1 SAs with each sub-server.
- An IKE gateway is configured with dead peer detection (DPD) for each sub-server.
- The server cluster role is `root-server` and each sub-server is configured as an IKE gateway for the server cluster.

The root-server should be configured to support chassis cluster operation. In the example, redundant Ethernet interfaces on the root-server connect to each of the sub-servers in the server cluster; the entire chassis cluster configuration is not shown.

On each sub-server:

- Two IKE policies are configured: `RootSrv` is used to establish a Phase 1 SA with the root-server, and `GMs` is used to establish Phase 1 SAs with each group member.

  Preshared keys are used to secure the Phase 1 SAs between the root-server and the sub-servers and between the sub-servers and the group members. Ensure that the preshared keys used are strong keys. On the sub-servers, the preshared key configured for the IKE policy `RootSrv` must match the preshared key configured on the root-server, and the preshared key configured for the IKE policy `GMs` must match the preshared key configured on the group members.

- An IKE gateway is configured with DPD for the root-server. In addition, an IKE gateway is configured for each group member.
- The server cluster role is `sub-server` and the root-server is configured as the IKE gateway for the server cluster.

On each group member:

- The IKE policy `SubSrv` is used to establish Phase 1 SAs with the sub-servers.
- The IKE gateway configuration includes the addresses for the sub-servers.

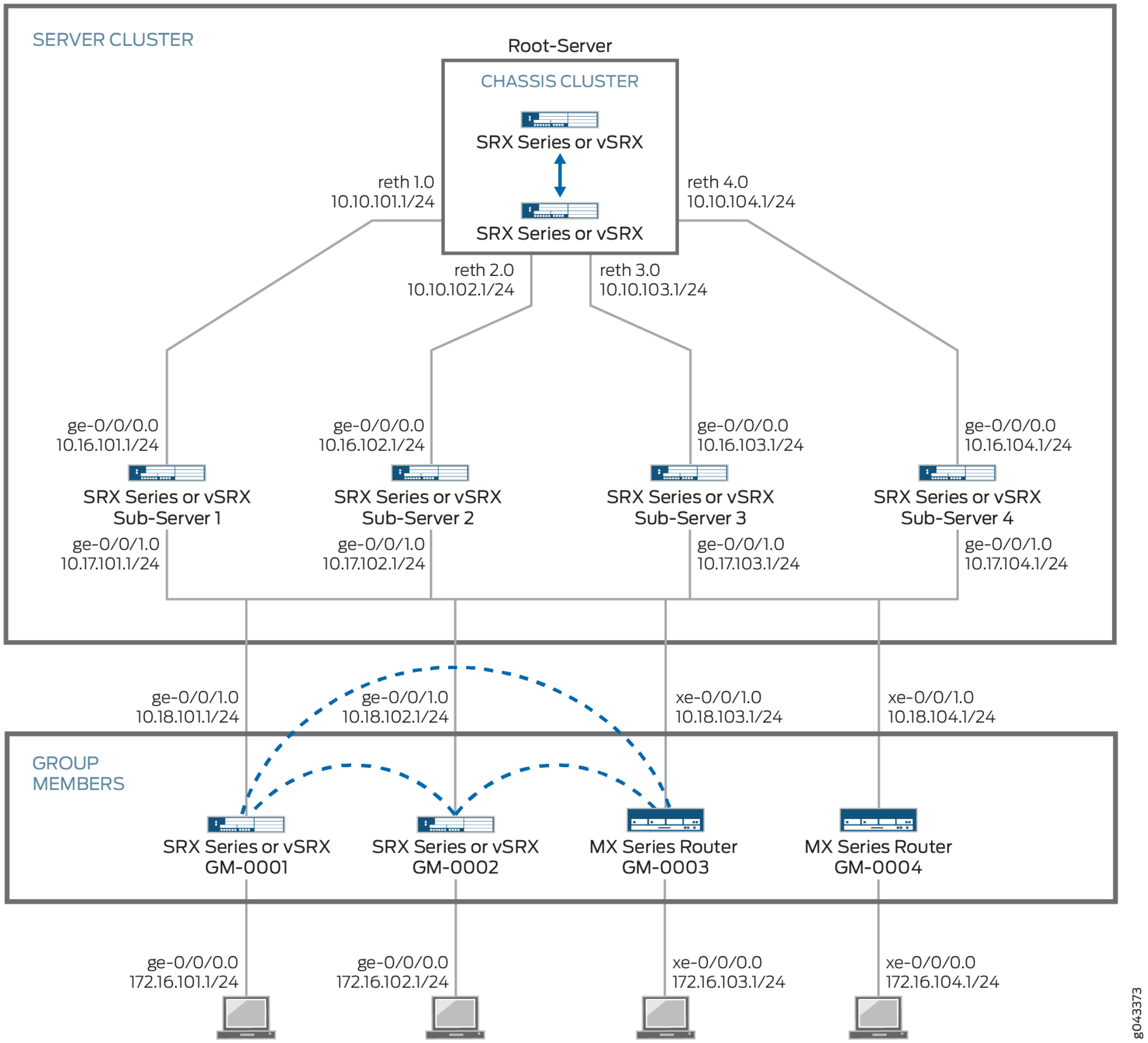
On SRX Series Firewalls or vSRX Virtual Firewall group members, an IPsec policy is configured for the group with the LAN zone as the from-zone (incoming traffic) and the WAN zone as the to-zone (outgoing traffic). A security policy is also needed to allow traffic between the LAN and WAN zones.

The same group identifier must be configured on both the group server and the group members. In this example, the group name is GROUP_ID-0001 and the group identifier is 1. The group policy configured on the server specifies that the SA and key are applied to traffic between subnetworks in the 172.16.0.0/12 range.

Topology

Figure 57 on page 835 shows the Juniper Networks devices to be configured for this example.

Figure 57: Group VPNv2 Server Cluster with SRX Series or vSRX Virtual Firewall and MX Series Members



Configuration

IN THIS SECTION

Configuring the Root-Server

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 description To_SubSrv01
set interfaces reth1 unit 0 family inet address 10.10.101.1/24
set interfaces reth2 redundant-ether-options redundancy-group 1
set interfaces reth2 unit 0 description To_SubSrv02
set interfaces reth2 unit 0 family inet address 10.10.102.1/24
set interfaces reth3 redundant-ether-options redundancy-group 1
set interfaces reth3 unit 0 description To_SubSrv03
set interfaces reth3 unit 0 family inet address 10.10.103.1/24
set interfaces reth4 redundant-ether-options redundancy-group 1
set interfaces reth4 unit 0 description To_SubSrv04
set interfaces reth4 unit 0 family inet address 10.10.104.1/24
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces reth1.0
set security zones security-zone GROUPVPN interfaces reth2.0
set security zones security-zone GROUPVPN interfaces reth3.0
set security zones security-zone GROUPVPN interfaces reth4.0
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set chassis cluster reth-count 5
set chassis cluster redundancy-group 1 node 0 priority 254
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-cbc
set security group-vpn server ike policy SubSrv mode main
set security group-vpn server ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy SubSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike gateway SubSrv01 ike-policy SubSrv
set security group-vpn server ike gateway SubSrv01 address 10.16.101.1
set security group-vpn server ike gateway SubSrv01 dead-peer-detection always-send
set security group-vpn server ike gateway SubSrv01 local-address 10.10.101.1
set security group-vpn server ike gateway SubSrv02 ike-policy SubSrv
set security group-vpn server ike gateway SubSrv02 address 10.16.102.1
set security group-vpn server ike gateway SubSrv02 dead-peer-detection always-send
set security group-vpn server ike gateway SubSrv02 local-address 10.10.102.1
set security group-vpn server ike gateway SubSrv03 ike-policy SubSrv
set security group-vpn server ike gateway SubSrv03 address 10.16.103.1
set security group-vpn server ike gateway SubSrv03 dead-peer-detection always-send
set security group-vpn server ike gateway SubSrv03 local-address 10.10.103.1
set security group-vpn server ike gateway SubSrv04 ike-policy SubSrv
set security group-vpn server ike gateway SubSrv04 address 10.16.104.1
set security group-vpn server ike gateway SubSrv04 dead-peer-detection always-send
set security group-vpn server ike gateway SubSrv04 local-address 10.10.104.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 server-cluster server-role root-server
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway SubSrv01
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway SubSrv02
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway SubSrv03
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway SubSrv04
set security group-vpn server group GROUP_ID-0001 server-cluster retransmission-period 10
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type
unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-
algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds
7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm
sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-
L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source
172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1
destination 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol
0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the root-server:

1. Configure security zones and security policies.

```
[edit interfaces]
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 description To_SubSrv01
user@host# set reth1 unit 0 family inet address 10.10.101.1/24
user@host# set reth2 redundant-ether-options redundancy-group 1
user@host# set reth2 unit 0 description To_SubSrv02
user@host# set reth2 unit 0 family inet address 10.10.102.1/24
user@host# set reth3 redundant-ether-options redundancy-group 1
user@host# set reth3 unit 0 description To_SubSrv03
```

```
user@host# set reth3 unit 0 family inet address 10.10.103.1/24
user@host# set reth4 redundant-ether-options redundancy-group 1
user@host# set reth4 unit 0 description To_SubSrv04
user@host# set reth4 unit 0 family inet address 10.10.104.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces reth1.0
user@host# set interfaces reth2.0
user@host# set interfaces reth3.0
user@host# set interfaces reth4.0
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit security policies]
user@host# set default-policy deny-all
```

2. Configure the chassis cluster.

```
[edit chassis cluster]
user@host# set reth-count 5
user@host# set redundancy-group 1 node 0 priority 254
user@host# set redundancy-group 1 node 1 priority 1
user@host# set redundancy-group 0 node 0 priority 254
user@host# set redundancy-group 0 node 1 priority 1
```

3. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy SubSrv]
```

```
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike gateway SubSrv01]
user@host# set ike-policy SubSrv
user@host# set address 10.16.101.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.10.101.1
[edit security group-vpn server ike gateway SubSrv02]
user@host# set ike-policy SubSrv
user@host# set address 10.16.102.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.10.102.1
[edit security group-vpn server ike gateway SubSrv03]
user@host# set ike-policy SubSrv
user@host# set address 10.16.103.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.10.103.1
[edit security group-vpn server ike gateway SubSrv04]
user@host# set ike-policy SubSrv
user@host# set address 10.16.104.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.10.104.1
```

4. Configure the IPsec SA.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600
```

5. Configure the VPN group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set server-cluster server-role root-server
user@host# set server-cluster ike-gateway SubSrv01
user@host# set server-cluster ike-gateway SubSrv02
user@host# set server-cluster ike-gateway SubSrv03
user@host# set server-cluster ike-gateway SubSrv04
```

```
user@host# set server-cluster retransmission-period 10
user@host# set anti-replay-time-window 1000
user@host# set server-member-communication communication-type unicast
user@host# set server-member-communication encryption-algorithm aes-256-cbc
user@host# set server-member-communication lifetime-seconds 7200
user@host# set server-member-communication sig-hash-algorithm sha-256
```

6. Configure the group policy.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
user@host# set ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show chassis cluster`, and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        description To_SubSrv01;
        family inet {
            address 10.10.101.1/24;
        }
    }
}
reth2 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        description To_SubSrv02;
        family inet {
            address 10.10.102.1/24;
        }
    }
}
reth3 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        description To_SubSrv03;
        family inet {
            address 10.10.103.1/24;
        }
    }
}
reth4 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        description To_SubSrv04;
        family inet {
            address 10.10.104.1/24;
        }
    }
}
[edit]
user@host# show chassis cluster
reth-count 5;
redundancy-group 1 {
    node 0 priority 254;
    node 1 priority 1;
}
redundancy-group 0 {
    node 0 priority 254;
    node 1 priority 1;
}
[edit]
user@host# show security
group-vpn {
    server {
        ike {
```

```
        proposal PSK-SHA256-DH14-AES256 {
            authentication-method pre-shared-keys;
            authentication-algorithm sha-256;
            dh-group group14;
            encryption-algorithm aes-256-cbc;
        }
        policy SubSrv {
            mode main;
            proposals PSK-SHA256-DH14-AES256;
            pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
        }
        gateway SubSrv01 {
            ike-policy SubSrv;
            address 10.16.101.1;
            dead-peer-detection always-send;
            local-address 10.10.101.1;
        }
        gateway SubSrv02 {
            ike-policy SubSrv;
            address 10.16.102.1;
            dead-peer-detection always-send;
            local-address 10.10.102.1;
        }
        gateway SubSrv03 {
            ike-policy SubSrv;
            address 10.16.103.1;
            dead-peer-detection always-send;
            local-address 10.10.103.1;
        }
        gateway SubSrv04 {
            ike-policy SubSrv;
            address 10.16.104.1;
            dead-peer-detection always-send;
            local-address 10.10.104.1;
        }
    }
    ipsec {
        proposal AES256-SHA256-L3600 {
            authentication-algorithm hmac-sha-256-128;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 3600;
        }
    }
```

```
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            server-cluster {
                server-role root-server;
                ike-gateway SubSrv01;
                ike-gateway SubSrv02;
                ike-gateway SubSrv03;
                ike-gateway SubSrv04;
                retransmission-period 10;
            }
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            reth1.0;
            reth2.0;
            reth3.0;
            reth4.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Sub-Server 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_RootSrv
set interfaces ge-0/0/0 unit 0 family inet address 10.16.101.1/24
set interfaces ge-0/0/1 unit 0 description To_WAN
set interfaces ge-0/0/1 unit 0 family inet address 10.17.101.1/24
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/0.0
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn server ike policy RootSrv mode main
set security group-vpn server ike policy RootSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy RootSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn server ike gateway RootSrv ike-policy RootSrv
set security group-vpn server ike gateway RootSrv address 10.10.101.1
set security group-vpn server ike gateway RootSrv dead-peer-detection always-send
set security group-vpn server ike gateway RootSrv local-address 10.16.101.1
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.17.101.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.17.101.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.17.101.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
set security group-vpn server ike gateway GM-0004 local-address 10.17.101.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-
```

```
sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 server-cluster server-role sub-server
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway RootSrv
set security group-vpn server group GROUP_ID-0001 server-cluster retransmission-period 10
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type
unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-
algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds
7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm
sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-
L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source
172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1
destination 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol
0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the sub-server in the Group VPNv2 server cluster:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_RootSrv
user@host# set ge-0/0/0 unit 0 family inet address 10.16.101.1/24
```

```
user@host# set ge-0/0/1 unit 0 description To_WAN
user@host# set ge-0/0/1 unit 0 family inet address 10.17.101.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
user@host# set interfaces ge-0/0/1.0
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit security policies]
user@host# set default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy RootSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike policy GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn server ike gateway RootSrv]
user@host# set ike-policy RootSrv
user@host# set address 10.10.101.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.16.101.1
[edit security group-vpn server ike gateway GM-0001]
user@host# set ike-policy GMs
```

```
user@host# set address 10.18.101.1
user@host# set local-address 10.17.101.1
[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.17.101.1
[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.17.101.1
[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.17.101.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600
```

4. Configure the VPN group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set server-cluster server-role sub-server
user@host# set server-cluster ike-gateway RootSrv
user@host# set server-cluster retransmission-period 10
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
user@host# set server-member-communication communication-type unicast
user@host# set server-member-communication encryption-algorithm aes-256-cbc
user@host# set server-member-communication lifetime-seconds 7200
user@host# set server-member-communication sig-hash-algorithm sha-256
user@host# set ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
```

5. Configure the group policy.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_RootSrv;
        family inet {
            address 10.16.101.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_WAN;
        family inet {
            address 10.17.101.1/24;
        }
    }
}
[edit]
user@host# show security
group-vpn {
    server {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
                encryption-algorithm aes-256-cbc;
            }
            policy RootSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            policy GMs {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway RootSrv {
                ike-policy RootSrv;
                address 10.10.101.1;
                dead-peer-detection always-send;
                local-address 10.16.101.1;
            }
            gateway GM-0001 {
                ike-policy GMs;
                address 10.18.101.1;
                local-address 10.17.101.1;
            }
            gateway GM-0002 {
                ike-policy GMs;
                address 10.18.102.1;
                local-address 10.17.101.1;
            }
            gateway GM-0003 {
                ike-policy GMs;
                address 10.18.103.1;
                local-address 10.17.101.1;
            }
            gateway GM-0004 {
                ike-policy GMs;
                address 10.18.104.1;
                local-address 10.17.101.1;
            }
        }
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
                lifetime-seconds 3600;
            }
        }
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            server-cluster {
                server-role sub-server;
                ike-gateway RootSrv;
                retransmission-period 10;
            }
            ike-gateway GM-0001;
            ike-gateway GM-0002;
            ike-gateway GM-0003;
            ike-gateway GM-0004;
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Sub-Server 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_RootSrv
set interfaces ge-0/0/0 unit 0 family inet address 10.16.102.1/24
set interfaces ge-0/0/1 unit 0 description To_WAN
set interfaces ge-0/0/1 unit 0 family inet address 10.17.102.1/24
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/0.0
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn server ike policy RootSrv mode main
set security group-vpn server ike policy RootSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy RootSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn server ike gateway RootSrv ike-policy RootSrv
set security group-vpn server ike gateway RootSrv address 10.10.102.1
set security group-vpn server ike gateway RootSrv dead-peer-detection always-send
set security group-vpn server ike gateway RootSrv local-address 10.16.102.1
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.17.102.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.17.102.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.17.102.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
set security group-vpn server ike gateway GM-0004 local-address 10.17.102.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 server-cluster server-role sub-server
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway RootSrv
set security group-vpn server group GROUP_ID-0001 server-cluster retransmission-period 10
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds 7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the sub-server in the Group VPNv2 server cluster:

1. Configure interfaces, security zones, and security policies.

   ```
   [edit interfaces]
   user@host# set ge-0/0/0 unit 0 description To_RootSrv
   ```

```
user@host# set ge-0/0/0 unit 0 family inet address 10.16.102.1/24
user@host# set ge-0/0/1 unit 0 description To_WAN
user@host# set ge-0/0/1 unit 0 family inet address 10.17.102.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
user@host# set interfaces ge-0/0/1.0
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit security policies]
user@host# set default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy RootSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike policy GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn server ike gateway RootSrv]
user@host# set ike-policy RootSrv
user@host# set address 10.10.102.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.16.102.1
[edit security group-vpn server ike gateway GM-0001]
```

```
user@host# set ike-policy GMs
user@host# set address 10.18.101.1
user@host# set local-address 10.17.102.1
[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.17.102.1
[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.17.102.1
[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.17.102.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600
```

4. Configure the VPN group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set server-cluster server-role sub-server
user@host# set server-cluster ike-gateway RootSrv
user@host# set server-cluster retransmission-period 10
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
user@host# set server-member-communication communication-type unicast
user@host# set server-member-communication encryption-algorithm aes-256-cbc
user@host# set server-member-communication lifetime-seconds 7200
user@host# set server-member-communication sig-hash-algorithm sha-256
```

5. Configure the group policy.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
user@host# set ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_RootSrv;
        family inet {
            address 10.16.102.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_WAN;
        family inet {
            address 10.17.102.1/24;
        }
    }
}
[edit]
user@host# show security
group-vpn {
    server {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
                encryption-algorithm aes-256-cbc;
            }
            policy RootSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            policy GMs {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway RootSrv {
                ike-policy RootSrv;
                address 10.10.102.1;
                dead-peer-detection always-send;
                local-address 10.16.102.1;
            }
            gateway GM-0001 {
                ike-policy GMs;
                address 10.18.101.1;
                local-address 10.17.102.1;
            }
            gateway GM-0002 {
                ike-policy GMs;
                address 10.18.102.1;
                local-address 10.17.102.1;
            }
            gateway GM-0003 {
                ike-policy GMs;
                address 10.18.103.1;
                local-address 10.17.102.1;
            }
            gateway GM-0004 {
                ike-policy GMs;
                address 10.18.104.1;
                local-address 10.17.102.1;
            }
        }
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
```

```
                lifetime-seconds 3600;
            }
        }
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            server-cluster {
                server-role sub-server;
                ike-gateway RootSrv;
                retransmission-period 10;
            }
            ike-gateway GM-0001;
            ike-gateway GM-0002;
            ike-gateway GM-0003;
            ike-gateway GM-0004;
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Sub-Server 3

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_RootSrv
set interfaces ge-0/0/0 unit 0 family inet address 10.16.103.1/24
set interfaces ge-0/0/1 unit 0 description To_WAN
set interfaces ge-0/0/1 unit 0 family inet address 10.17.103.1/24
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/0.0
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn server ike policy RootSrv mode main
set security group-vpn server ike policy RootSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy RootSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn server ike gateway RootSrv ike-policy RootSrv
set security group-vpn server ike gateway RootSrv address 10.10.103.1
set security group-vpn server ike gateway RootSrv dead-peer-detection always-send
set security group-vpn server ike gateway RootSrv local-address 10.16.103.1
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.17.103.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.17.103.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.17.103.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
set security group-vpn server ike gateway GM-0004 local-address 10.17.103.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 server-cluster server-role sub-server
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway RootSrv
set security group-vpn server group GROUP_ID-0001 server-cluster retransmission-period 10
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds 7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the sub-server in the Group VPNv2 server cluster:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_RootSrv
user@host# set ge-0/0/0 unit 0 family inet address 10.16.103.1/24
user@host# set ge-0/0/1 unit 0 description To_WAN
user@host# set ge-0/0/1 unit 0 family inet address 10.17.103.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
user@host# set interfaces ge-0/0/1.0
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit security policies]
user@host# set default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy RootSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike policy GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn server ike gateway RootSrv]
```

```
user@host# set ike-policy RootSrv
user@host# set address 10.10.103.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.16.103.1
[edit security group-vpn server ike gateway GM-0001]
user@host# set ike-policy GMs
user@host# set address 10.18.101.1
user@host# set local-address 10.17.103.1
[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.17.103.1
[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.17.103.1
[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.17.103.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600
```

4. Configure the VPN group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set server-cluster server-role sub-server
user@host# set server-cluster ike-gateway RootSrv
user@host# set server-cluster retransmission-period 10
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
```

```
user@host# set server-member-communication communication-type unicast
user@host# set server-member-communication encryption-algorithm aes-256-cbc
user@host# set server-member-communication lifetime-seconds 7200
user@host# set server-member-communication sig-hash-algorithm sha-256
```

5. Configure the group policy.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
user@host# set ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_RootSrv;
        family inet {
            address 10.16.103.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_WAN;
        family inet {
            address 10.17.103.1/24;
        }
    }
}
[edit]
user@host# show security
group-vpn {
    server {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
                encryption-algorithm aes-256-cbc;
            }
            policy RootSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            policy GMs {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway RootSrv {
                ike-policy RootSrv;
                address 10.10.103.1;
                dead-peer-detection always-send;
                local-address 10.16.103.1;
            }
            gateway GM-0001 {
                ike-policy GMs;
                address 10.18.101.1;
                local-address 10.17.103.1;
            }
            gateway GM-0002 {
                ike-policy GMs;
                address 10.18.102.1;
                local-address 10.17.103.1;
            }
            gateway GM-0003 {
                ike-policy GMs;
                address 10.18.103.1;
                local-address 10.17.103.1;
            }
            gateway GM-0004 {
                ike-policy GMs;
                address 10.18.104.1;
                local-address 10.17.103.1;
            }
        }
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
                lifetime-seconds 3600;
            }
        }
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            server-cluster {
                server-role sub-server;
                ike-gateway RootSrv;
                retransmission-period 10;
            }
            ike-gateway GM-0001;
            ike-gateway GM-0002;
            ike-gateway GM-0003;
            ike-gateway GM-0004;
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Sub-Server 4

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_RootSrv
set interfaces ge-0/0/0 unit 0 family inet address 10.16.104.1/24
set interfaces ge-0/0/1 unit 0 description To_WAN
set interfaces ge-0/0/1 unit 0 family inet address 10.17.104.1/24
set security zones security-zone GROUPVPN host-inbound-traffic system-services ike
set security zones security-zone GROUPVPN host-inbound-traffic system-services ssh
set security zones security-zone GROUPVPN host-inbound-traffic system-services ping
set security zones security-zone GROUPVPN interfaces ge-0/0/0.0
set security zones security-zone GROUPVPN interfaces ge-0/0/1.0
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn server ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn server ike policy RootSrv mode main
set security group-vpn server ike policy RootSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy RootSrv pre-shared-key ascii-text "$ABC123"
set security group-vpn server ike policy GMs mode main
set security group-vpn server ike policy GMs proposals PSK-SHA256-DH14-AES256
set security group-vpn server ike policy GMs pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn server ike gateway RootSrv ike-policy RootSrv
set security group-vpn server ike gateway RootSrv address 10.10.104.1
set security group-vpn server ike gateway RootSrv dead-peer-detection always-send
set security group-vpn server ike gateway RootSrv local-address 10.16.104.1
set security group-vpn server ike gateway GM-0001 ike-policy GMs
set security group-vpn server ike gateway GM-0001 address 10.18.101.1
set security group-vpn server ike gateway GM-0001 local-address 10.17.104.1
set security group-vpn server ike gateway GM-0002 ike-policy GMs
set security group-vpn server ike gateway GM-0002 address 10.18.102.1
set security group-vpn server ike gateway GM-0002 local-address 10.17.104.1
set security group-vpn server ike gateway GM-0003 ike-policy GMs
set security group-vpn server ike gateway GM-0003 address 10.18.103.1
set security group-vpn server ike gateway GM-0003 local-address 10.17.104.1
set security group-vpn server ike gateway GM-0004 ike-policy GMs
set security group-vpn server ike gateway GM-0004 address 10.18.104.1
set security group-vpn server ike gateway GM-0004 local-address 10.17.104.1
set security group-vpn server ipsec proposal AES256-SHA256-L3600 authentication-algorithm hmac-
sha-256-128
set security group-vpn server ipsec proposal AES256-SHA256-L3600 encryption-algorithm aes-256-cbc
set security group-vpn server ipsec proposal AES256-SHA256-L3600 lifetime-seconds 3600
set security group-vpn server group GROUP_ID-0001 group-id 1
set security group-vpn server group GROUP_ID-0001 member-threshold 2000
set security group-vpn server group GROUP_ID-0001 server-cluster server-role sub-server
set security group-vpn server group GROUP_ID-0001 server-cluster ike-gateway RootSrv
set security group-vpn server group GROUP_ID-0001 server-cluster retransmission-period 10
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0001
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0002
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0003
set security group-vpn server group GROUP_ID-0001 ike-gateway GM-0004
set security group-vpn server group GROUP_ID-0001 anti-replay-time-window 1000
set security group-vpn server group GROUP_ID-0001 server-member-communication communication-type
unicast
set security group-vpn server group GROUP_ID-0001 server-member-communication encryption-
algorithm aes-256-cbc
set security group-vpn server group GROUP_ID-0001 server-member-communication lifetime-seconds
7200
set security group-vpn server group GROUP_ID-0001 server-member-communication sig-hash-algorithm
sha-256
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-
L3600
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 source
172.16.0.0/12
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1
destination 172.16.0.0/12
```

```
set security group-vpn server group GROUP_ID-0001 ipsec-sa GROUP_ID-0001 match-policy 1 protocol
0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the sub-server in the Group VPNv2 server cluster:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_RootSrv
user@host# set ge-0/0/0 unit 0 family inet address 10.16.104.1/24
user@host# set ge-0/0/1 unit 0 description To_WAN
user@host# set ge-0/0/1 unit 0 family inet address 10.17.104.1/24
[edit security zones security-zone GROUPVPN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
user@host# set interfaces ge-0/0/1.0
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit security policies]
user@host# set default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn server ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
```

```
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn server ike policy RootSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123"
[edit security group-vpn server ike policy GMs]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn server ike gateway RootSrv]
user@host# set ike-policy RootSrv
user@host# set address 10.10.104.1
user@host# set dead-peer-detection always-send
user@host# set local-address 10.16.104.1
[edit security group-vpn server ike gateway GM-0001]
user@host# set ike-policy GMs
user@host# set address 10.18.101.1
user@host# set local-address 10.17.104.1
[edit security group-vpn server ike gateway GM-0002]
user@host# set ike-policy GMs
user@host# set address 10.18.102.1
user@host# set local-address 10.17.104.1
[edit security group-vpn server ike gateway GM-0003]
user@host# set ike-policy GMs
user@host# set address 10.18.103.1
user@host# set local-address 10.17.104.1
[edit security group-vpn server ike gateway GM-0004]
user@host# set ike-policy GMs
user@host# set address 10.18.104.1
user@host# set local-address 10.17.104.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn server ipsec proposal AES256-SHA256-L3600]
user@host# set authentication-algorithm hmac-sha-256-128
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 3600
```

4. Configure the VPN group.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set group-id 1
user@host# set member-threshold 2000
user@host# set server-cluster server-role sub-server
user@host# set server-cluster ike-gateway RootSrv
user@host# set server-cluster retransmission-period 10
user@host# set ike-gateway GM-0001
user@host# set ike-gateway GM-0002
user@host# set ike-gateway GM-0003
user@host# set ike-gateway GM-0004
user@host# set anti-replay-time-window 1000
user@host# set server-member-communication communication-type unicast
user@host# set server-member-communication encryption-algorithm aes-256-cbc
user@host# set server-member-communication lifetime-seconds 7200
user@host# set server-member-communication sig-hash-algorithm sha-256
```

5. Configure the group policy.

```
[edit security group-vpn server group GROUP_ID-0001]
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 source 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 destination 172.16.0.0/12
user@host# set ipsec-sa GROUP_ID-0001 match-policy 1 protocol 0
user@host# set ipsec-sa GROUP_ID-0001 proposal AES256-SHA256-L3600
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_RootSrv;
        family inet {
            address 10.16.104.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_WAN;
        family inet {
            address 10.17.104.1/24;
        }
    }
}
[edit]
user@host# show security
group-vpn {
    server {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                authentication-algorithm sha-256;
                dh-group group14;
                encryption-algorithm aes-256-cbc;
            }
            policy RootSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
            policy GMs {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway RootSrv {
                ike-policy RootSrv;
                address 10.10.104.1;
                dead-peer-detection always-send;
                local-address 10.16.104.1;
            }
            gateway GM-0001 {
                ike-policy GMs;
                address 10.18.101.1;
                local-address 10.17.104.1;
            }
            gateway GM-0002 {
                ike-policy GMs;
                address 10.18.102.1;
                local-address 10.17.104.1;
            }
            gateway GM-0003 {
                ike-policy GMs;
                address 10.18.103.1;
                local-address 10.17.104.1;
            }
            gateway GM-0004 {
                ike-policy GMs;
                address 10.18.104.1;
                local-address 10.17.104.1;
            }
        }
        ipsec {
            proposal AES256-SHA256-L3600 {
                authentication-algorithm hmac-sha-256-128;
                encryption-algorithm aes-256-cbc;
                lifetime-seconds 3600;
            }
        }
        group GROUP_ID-0001 {
            group-id 1;
            member-threshold 2000;
            server-cluster {
                server-role sub-server;
                ike-gateway RootSrv;
                retransmission-period 10;
            }
            ike-gateway GM-0001;
            ike-gateway GM-0002;
            ike-gateway GM-0003;
            ike-gateway GM-0004;
            anti-replay-time-window 1000;
            server-member-communication {
                communication-type unicast;
                lifetime-seconds 7200;
                encryption-algorithm aes-256-cbc;
                sig-hash-algorithm sha-256;
            }
            ipsec-sa GROUP_ID-0001 {
                proposal AES256-SHA256-L3600;
                match-policy 1 {
                    source 172.16.0.0/12;
                    destination 172.16.0.0/12;
                    protocol 0;
                }
            }
        }
    }
}
policies {
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone GROUPVPN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring GM-0001 (SRX Series Firewall or vSRX Virtual Firewall Instance)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_LAN
set interfaces ge-0/0/0 unit 0 family inet address 172.16.101.1/24
set interfaces ge-0/0/1 unit 0 description To_SubSrv
set interfaces ge-0/0/1 unit 0 family inet address 10.18.101.1/24
set security zones security-zone LAN host-inbound-traffic system-services ike
set security zones security-zone LAN host-inbound-traffic system-services ssh
set security zones security-zone LAN host-inbound-traffic system-services ping
set security zones security-zone LAN interfaces ge-0/0/0.0
set security zones security-zone WAN host-inbound-traffic system-services ike
set security zones security-zone WAN host-inbound-traffic system-services ssh
set security zones security-zone WAN host-inbound-traffic system-services ping
set security zones security-zone WAN interfaces ge-0/0/1.0
set security address-book global address 172.16.0.0/12 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match application any
set security policies from-zone LAN to-zone WAN policy 1 then permit
set security policies from-zone LAN to-zone WAN policy 1 then log session-init
set security policies from-zone WAN to-zone LAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match application any
set security policies from-zone WAN to-zone LAN policy 1 then permit
set security policies from-zone WAN to-zone LAN policy 1 then log session-init
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy SubSrv mode main
set security group-vpn member ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy SubSrv pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn member ike gateway SubSrv ike-policy SubSrv
set security group-vpn member ike gateway SubSrv server-address 10.17.101.1
set security group-vpn member ike gateway SubSrv server-address 10.17.102.1
set security group-vpn member ike gateway SubSrv server-address 10.17.103.1
set security group-vpn member ike gateway SubSrv server-address 10.17.104.1
set security group-vpn member ike gateway SubSrv local-address 10.18.101.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway SubSrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group-vpn-external-interface ge-0/0/1.0
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 recovery-probe
set security ipsec-policy from-zone LAN to-zone WAN ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure interfaces, security zones, and security policies.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 description To_LAN
user@host# set ge-0/0/0 unit 0 family inet address 172.16.101.1/24
user@host# set ge-0/0/1 unit 0 description To_SubSrv
user@host# set ge-0/0/1 unit 0 family inet address 10.18.101.1/24
[edit security zones security-zone LAN]
```

```
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
[edit security zones security-zone WAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0
[edit security]
user@host# set address-book global address 172.16.0.0/12 172.16.0.0/12
[edit security policies from-zone LAN to-zone WAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set policy 1 then log session-init
[edit security policies from-zone WAN to-zone LAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set policy 1 then log session-init
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit]
user@host# set security policies default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
```

```
[edit security group-vpn member ike policy SubSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn member ike gateway SubSrv]
user@host# set ike-policy SubSrv
user@host# set server-address 10.17.101.1
user@host# set server-address 10.17.102.1
user@host# set server-address 10.17.103.1
user@host# set server-address 10.17.104.1
user@host# set local-address 10.18.101.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway SubSrv
user@host# set group-vpn-external-interface ge-0/0/1.0
user@host# set group 1
user@host# set recovery-probe
```

4. Configure the IPsec policy.

```
[edit security ipsec-policy from-zone LAN to-zone WAN]
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_LAN;
        family inet {
            address 172.16.101.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        description To_SubSrv;
        family inet {
            address 10.18.101.1/24;
        }
    }
}
[edit]
user@host# show security
address-book {
    global {
        address 172.16.0.0/12 172.16.0.0/12;
    }
}
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy SubSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway SubSrv {
                ike-policy SubSrv;
                server-address [ 10.17.101.1 10.17.102.1 10.17.103.1 10.17.104.1 ];
                local-address 10.18.101.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway SubSrv;
                group-vpn-external-interface ge-0/0/1.0;
                group 1;
                recovery-probe;
            }
        }
    }
}
ipsec-policy {
    from-zone LAN to-zone WAN {
        ipsec-group-vpn GROUP_ID-0001;
    }
}
policies {
    from-zone LAN to-zone WAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    from-zone WAN to-zone LAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone LAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone WAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring GM-0002 (SRX Series Firewall or vSRX Virtual Firewall Instance)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 description To_LAN
set interfaces ge-0/0/0 unit 0 family inet address 172.16.102.1/24
set interfaces ge-0/0/1 unit 0 description To_SubSrv
set interfaces ge-0/0/1 unit 0 family inet address 10.18.102.1/24
set security zones security-zone LAN host-inbound-traffic system-services ike
set security zones security-zone LAN host-inbound-traffic system-services ssh
set security zones security-zone LAN host-inbound-traffic system-services ping
set security zones security-zone LAN interfaces ge-0/0/0.0
set security zones security-zone WAN host-inbound-traffic system-services ike
set security zones security-zone WAN host-inbound-traffic system-services ssh
set security zones security-zone WAN host-inbound-traffic system-services ping
set security zones security-zone WAN interfaces ge-0/0/1.0
set security address-book global address 172.16.0.0/12 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone LAN to-zone WAN policy 1 match application any
set security policies from-zone LAN to-zone WAN policy 1 then permit
set security policies from-zone LAN to-zone WAN policy 1 then log session-init
set security policies from-zone WAN to-zone LAN policy 1 match source-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match destination-address 172.16.0.0/12
set security policies from-zone WAN to-zone LAN policy 1 match application any
set security policies from-zone WAN to-zone LAN policy 1 then permit
set security policies from-zone WAN to-zone LAN policy 1 then log session-init
set security policies global policy 1000 match source-address any
set security policies global policy 1000 match destination-address any
set security policies global policy 1000 match application any
set security policies global policy 1000 match from-zone any
set security policies global policy 1000 match to-zone any
set security policies global policy 1000 then deny
set security policies global policy 1000 then log session-init
set security policies global policy 1000 then count
set security policies default-policy deny-all
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy SubSrv mode main
set security group-vpn member ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy SubSrv pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn member ike gateway SubSrv ike-policy SubSrv
set security group-vpn member ike gateway SubSrv server-address 10.17.101.1
set security group-vpn member ike gateway SubSrv server-address 10.17.102.1
set security group-vpn member ike gateway SubSrv server-address 10.17.103.1
set security group-vpn member ike gateway SubSrv server-address 10.17.104.1
set security group-vpn member ike gateway SubSrv local-address 10.18.102.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway SubSrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group-vpn-external-interface ge-0/0/1.0
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 recovery-probe
set security ipsec-policy from-zone LAN to-zone WAN ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure interfaces, security zones, and security policies.

   ```
   [edit interfaces]
   user@host# set ge-0/0/0 unit 0 description To_LAN
   user@host# set ge-0/0/0 unit 0 family inet address 172.16.102.1/24
   user@host# set ge-0/0/1 unit 0 description To_SubSrv
   user@host# set ge-0/0/1 unit 0 family inet address 10.18.102.1/24
   [edit security zones security-zone LAN]
   user@host# set host-inbound-traffic system-services ike
   user@host# set host-inbound-traffic system-services ssh
   ```

```
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/0.0
[edit security zones security-zone WAN]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services ssh
user@host# set host-inbound-traffic system-services ping
user@host# set interfaces ge-0/0/1.0
[edit security]
user@host# set address-book global address 172.16.0.0/12 172.16.0.0/12
[edit security policies from-zone LAN to-zone WAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set policy 1 then log session-init
[edit security policies from-zone WAN to-zone LAN]
user@host# set policy 1 match source-address 172.16.0.0/12
user@host# set policy 1 match destination-address 172.16.0.0/12
user@host# set policy 1 match application any
user@host# set policy 1 then permit
user@host# set policy 1 then log session-init
[edit security policies global]
user@host# set policy 1000 match source-address any
user@host# set policy 1000 match destination-address any
user@host# set policy 1000 match application any
user@host# set policy 1000 match from-zone any
user@host# set policy 1000 match to-zone any
user@host# set policy 1000 then deny
user@host# set policy 1000 then log session-init
user@host# set policy 1000 then count
[edit]
user@host# set security policies default-policy deny-all
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy SubSrv]
user@host# set mode main
```

```
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn member ike gateway SubSrv]
user@host# set ike-policy SubSrv
user@host# set server-address 10.17.101.1
user@host# set server-address 10.17.102.1
user@host# set server-address 10.17.103.1
user@host# set server-address 10.17.104.1
user@host# set local-address 10.18.102.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway SubSrv
user@host# set group-vpn-external-interface ge-0/0/1.0
user@host# set group 1
user@host# set recovery-probe
```

4. Configure the IPsec policy.

```
[edit security ipsec-policy from-zone LAN to-zone WAN]
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show security` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        description To_LAN;
        family inet {
            address 172.16.102.1/24;
        }
    }
}
```

```
ge-0/0/1 {
    unit 0 {
        description To_SubSrv;
        family inet {
            address 10.18.102.1/24;
        }
    }
}
[edit]
user@host# show security
address-book {
    global {
        address 172.16.0.0/12 172.16.0.0/12;
    }
}
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy SubSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway SubSrv {
                ike-policy SubSrv;
                server-address [ 10.17.101.1 10.17.102.1 10.17.103.1 10.17.104.1 ];
                local-address 10.18.102.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway SubSrv;
                group-vpn-external-interface ge-0/0/1.0;
                group 1;
                recovery-probe;
            }
        }
    }
}
ipsec-policy {
    from-zone LAN to-zone WAN {
        ipsec-group-vpn GROUP_ID-0001;
    }
}
policies {
    from-zone LAN to-zone WAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    from-zone WAN to-zone LAN {
        policy 1 {
            match {
                source-address 172.16.0.0/12;
                destination-address 172.16.0.0/12;
                application any;
            }
            then {
                permit;
                log {
                    session-init;
                }
            }
        }
    }
    global {
        policy 1000 {
            match {
                source-address any;
                destination-address any;
                application any;
                from-zone any;
                to-zone any;
            }
            then {
                deny;
                log {
                    session-init;
                }
                count;
            }
        }
    }
    default-policy {
        deny-all;
    }
}
zones {
    security-zone LAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone WAN {
        host-inbound-traffic {
            system-services {
                ike;
                ssh;
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring GM-0003 (MX Series Device)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet address 10.18.103.1/24
set interfaces xe-0/0/2 unit 0 family inet address 172.16.103.1/24
set interfaces ms-0/2/0 unit 0 family inet
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy SubSrv mode main
set security group-vpn member ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy SubSrv pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn member ike gateway SubSrv ike-policy SubSrv
set security group-vpn member ike gateway SubSrv server-address 10.17.101.1
set security group-vpn member ike gateway SubSrv server-address 10.17.102.1
set security group-vpn member ike gateway SubSrv server-address 10.17.103.1
set security group-vpn member ike gateway SubSrv server-address 10.17.104.1
set security group-vpn member ike gateway SubSrv local-address 10.18.103.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway SubSrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 match-direction output
set security group-vpn member ipsec vpn GROUP_ID-0001 tunnel-mtu 1400
set security group-vpn member ipsec vpn GROUP_ID-0001 df-bit clear
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
```

```
10.17.101.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.102.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.103.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.104.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.101.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.102.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.103.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.104.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from source-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from destination-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 then service
set services service-set GROUP_ID-0001 interface-service service-interface ms-0/2/0.0
set services service-set GROUP_ID-0001 ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet address 10.18.103.1/24
```

```
user@host# set xe-0/0/2 unit 0 family inet address 172.16.103.1/24
user@host# set ms-0/2/0 unit 0 family inet
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy SubSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn member ike gateway SubSrv]
user@host# set ike-policy SubSrv
user@host# set server-address 10.17.101.1
user@host# set server-address 10.17.102.1
user@host# set server-address 10.17.103.1
user@host# set server-address 10.17.104.1
user@host# set local-address 10.18.103.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway SubSrv
user@host# set group 1
user@host# set match-direction output
user@host# set tunnel-mtu 1400
user@host# set df-bit clear
```

4. Configure the service filter.

```
[edit firewall family inet service-filter GroupVPN-KS]
user@host# set term inbound-ks from source-address 10.17.101.1/32
user@host# set term inbound-ks from source-address 10.17.102.1/32
user@host# set term inbound-ks from source-address 10.17.103.1/32
user@host# set term inbound-ks from source-address 10.17.104.1/32
user@host# set term inbound-ks then skip
user@host# set term outbound-ks from destination-address 10.17.101.1/32
```

```
user@host# set term outbound-ks from destination-address 10.17.102.1/32
user@host# set term outbound-ks from destination-address 10.17.103.1/32
user@host# set term outbound-ks from destination-address 10.17.104.1/32
user@host# set term outbound-ks then skip
user@host# set term GROUP_ID-0001 from source-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 from destination-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 then service
```

5. Configure the service set.

```
[edit services service-set GROUP_ID-0001]
user@host# set interface-service service-interface ms-0/2/0.0
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security`, `show services`, and `show firewall` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
xe-0/0/1 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
                output {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
            }
            address 10.18.103.1/24;
        }
    }
}
xe-0/0/2 {
    unit 0 {
        family inet {
            address 172.16.103.1/24;
        }
    }
}
ms-0/2/0 {
    unit 0 {
        family inet;
    }
}
[edit]
user@host# show security
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy SubSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text "$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway SubSrv {
                ike-policy SubSrv;
                server-address [ 10.17.101.1 10.17.102.1 10.17.103.1 10.17.104.1 ];
                local-address 10.18.103.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway SubSrv;
                group 1;
                match-direction output;
                tunnel-mtu 1400;
                df-bit clear;
            }
        }
    }
}
[edit]
```

```
user@host# show services
service-set GROUP_ID-0001 {
    interface-service {
        service-interface ms-0/2/0.0;
    }
    ipsec-group-vpn GROUP_ID-0001;
}
[edit]
user@host# show firewall
family inet {
    service-filter GroupVPN-KS {
        term inbound-ks {
            from {
                source-address {
                    10.17.101.1/32;
                    10.17.102.1/32;
                    10.17.103.1/32;
                    10.17.104.1/32;
                }
            }
            then skip;
        }
        term outbound-ks {
            from {
                destination-address {
                    10.17.101.1/32;
                    10.17.102.1/32;
                    10.17.103.1/32;
                    10.17.104.1/32;
                }
            }
            then skip;
        }
        term GROUP_ID-0001 {
            from {
                source-address {
                    172.16.0.0/12;
                }
                destination-address {
                    172.16.0.0/12;
                }
            }
            then service;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring GM-0004 (MX Series Device)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
set interfaces xe-0/0/1 unit 0 family inet address 10.18.104.1/24
set interfaces xe-0/0/2 unit 0 family inet address 172.16.104.1/24
set interfaces ms-0/2/0 unit 0 family inet
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-method pre-
shared-keys
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 dh-group group14
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 authentication-algorithm
sha-256
set security group-vpn member ike proposal PSK-SHA256-DH14-AES256 encryption-algorithm aes-256-
cbc
set security group-vpn member ike policy SubSrv mode main
set security group-vpn member ike policy SubSrv proposals PSK-SHA256-DH14-AES256
set security group-vpn member ike policy SubSrv pre-shared-key ascii-text "$ABC123$ABC123"
set security group-vpn member ike gateway SubSrv ike-policy SubSrv
set security group-vpn member ike gateway SubSrv server-address 10.17.101.1
set security group-vpn member ike gateway SubSrv server-address 10.17.102.1
set security group-vpn member ike gateway SubSrv server-address 10.17.103.1
set security group-vpn member ike gateway SubSrv server-address 10.17.104.1
set security group-vpn member ike gateway SubSrv local-address 10.18.104.1
set security group-vpn member ipsec vpn GROUP_ID-0001 ike-gateway SubSrv
set security group-vpn member ipsec vpn GROUP_ID-0001 group 1
set security group-vpn member ipsec vpn GROUP_ID-0001 match-direction output
set security group-vpn member ipsec vpn GROUP_ID-0001 tunnel-mtu 1400
set security group-vpn member ipsec vpn GROUP_ID-0001 df-bit clear
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.101.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.102.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.103.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks from source-address
10.17.104.1/32
set firewall family inet service-filter GroupVPN-KS term inbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.101.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.102.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.103.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks from destination-address
10.17.104.1/32
set firewall family inet service-filter GroupVPN-KS term outbound-ks then skip
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from source-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 from destination-address
172.16.0.0/12
set firewall family inet service-filter GroupVPN-KS term GROUP_ID-0001 then service
set services service-set GROUP_ID-0001 interface-service service-interface ms-0/2/0.0
set services service-set GROUP_ID-0001 ipsec-group-vpn GROUP_ID-0001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the Group VPNv2 member:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set xe-0/0/1 unit 0 family inet service input service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet service output service-set GROUP_ID-0001 service-
filter GroupVPN-KS
user@host# set xe-0/0/1 unit 0 family inet address 10.18.104.1/24
```

```
user@host# set xe-0/0/2 unit 0 family inet address 172.16.104.1/24
user@host# set ms-0/2/0 unit 0 family inet
```

2. Configure the IKE proposal, policy, and gateway.

```
[edit security group-vpn member ike proposal PSK-SHA256-DH14-AES256]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group14
user@host# set authentication-algorithm sha-256
user@host# set encryption-algorithm aes-256-cbc
[edit security group-vpn member ike policy SubSrv]
user@host# set mode main
user@host# set proposals PSK-SHA256-DH14-AES256
user@host# set pre-shared-key ascii-text "$ABC123$ABC123"
[edit security group-vpn member ike gateway SubSrv]
user@host# set ike-policy SubSrv
user@host# set server-address 10.17.101.1
user@host# set server-address 10.17.102.1
user@host# set server-address 10.17.103.1
user@host# set server-address 10.17.104.1
user@host# set local-address 10.18.104.1
```

3. Configure the IPsec SA.

```
[edit security group-vpn member ipsec vpn GROUP_ID-0001]
user@host# set ike-gateway SubSrv
user@host# set group 1
user@host# set match-direction output
user@host# set tunnel-mtu 1400
user@host# set df-bit clear
```

4. Configure the service filter.

```
[edit firewall family inet service-filter GroupVPN-KS]
user@host# set term inbound-ks from source-address 10.17.101.1/32
user@host# set term inbound-ks from source-address 10.17.102.1/32
user@host# set term inbound-ks from source-address 10.17.103.1/32
user@host# set term inbound-ks from source-address 10.17.104.1/32
user@host# set term inbound-ks then skip
user@host# set term outbound-ks from destination-address 10.17.101.1/32
```

```
user@host# set term outbound-ks from destination-address 10.17.102.1/32
user@host# set term outbound-ks from destination-address 10.17.103.1/32
user@host# set term outbound-ks from destination-address 10.17.104.1/32
user@host# set term outbound-ks then skip
user@host# set term GROUP_ID-0001 from source-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 from destination-address 172.16.0.0/12
user@host# set term GROUP_ID-0001 then service
```

5. Configure the service set.

```
[edit services service-set GROUP_ID-0001]
user@host# set interface-service service-interface ms-0/2/0.0
user@host# set ipsec-group-vpn GROUP_ID-0001
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security`, `show services`, and `show firewall` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
xe-0/0/1 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
                output {
                    service-set GROUP_ID-0001 service-filter GroupVPN-KS;
                }
            }
            address 10.18.104.1/24;
        }
    }
}
xe-0/0/2 {
    unit 0 {
        family inet {
            address 172.16.104.1/24;
        }
    }
}
ms-0/2/0 {
    unit 0 {
        family inet;
    }
}
[edit]
user@host# show security
group-vpn {
    member {
        ike {
            proposal PSK-SHA256-DH14-AES256 {
                authentication-method pre-shared-keys;
                dh-group group14;
                authentication-algorithm sha-256;
                encryption-algorithm aes-256-cbc;
            }
            policy SubSrv {
                mode main;
                proposals PSK-SHA256-DH14-AES256;
                pre-shared-key ascii-text ""$ABC123$ABC123"; ## SECRET-DATA
            }
            gateway SubSrv {
                ike-policy SubSrv;
                server-address [ 10.17.101.1 10.17.102.1 10.17.103.1 10.17.104.1 ];
                local-address 10.18.104.1;
            }
        }
        ipsec {
            vpn GROUP_ID-0001 {
                ike-gateway SubSrv;
                group 1;
                match-direction output;
                tunnel-mtu 1400;
                df-bit clear;
            }
        }
    }
}
[edit]
```

```
user@host# show services
service-set GROUP_ID-0001 {
    interface-service {
        service-interface ms-0/2/0.0;
    }
    ipsec-group-vpn GROUP_ID-0001;
}
[edit]
user@host# show firewall
family inet {
    service-filter GroupVPN-KS {
        term inbound-ks {
            from {
                source-address {
                    10.17.101.1/32;
                    10.17.102.1/32;
                    10.17.103.1/32;
                    10.17.104.1/32;
                }
            }
            then skip;
        }
        term outbound-ks {
            from {
                destination-address {
                    10.17.101.1/32;
                    10.17.102.1/32;
                    10.17.103.1/32;
                    10.17.104.1/32;
                }
            }
            then skip;
        }
        term GROUP_ID-0001 {
            from {
                source-address {
                    172.16.0.0/12;
                }
                destination-address {
                    172.16.0.0/12;
                }
            }
            then service;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Server Cluster Operation

Purpose

Verify that devices in the server cluster recognize peer servers in the group. Ensure that the servers are active and roles in the cluster are properly assigned.

Action

From operational mode, enter the `show security group-vpn server server-cluster`, `show security group-vpn server server-cluster detail`, and `show security group-vpn server statistics` commands on the root-server.

```
user@RootSrv> show security group-vpn server server-cluster
Group: GROUP_ID-0001, Group Id: 1
Role: Root-server, Version Number: 2,
  Peer Gateway                    Peer IP         Role                    Status
  SubSrv01                        10.16.101.1   Sub-server              Active
  SubSrv02                        10.16.102.1   Sub-server              Active
  SubSrv03                         10.16.103.1   Sub-server              Active
  SubSrv04                         10.16.104.1   Sub-server              Active
```

```
user@RootSrv> show security group-vpn server server-cluster detail
Group: GROUP_ID-0001, Group Id: 1
Role: Root-server, Version Number: 2

Peer gateway: SubSrv01
  Peer IP: 10.16.101.1, Local IP: 10.10.101.1, VR: default
  Role: Sub-server, Status: Active
  CLUSTER-INIT send:                 0
  CLUSTER-INIT recv:                 1
  CLUSTER-INIT success:              1
  CLUSTER-INIT fail:                 0
  CLUSTER-INIT dup:                  0
  CLUSTER-INIT abort:                0
  CLUSTER-INIT timeout:              0
  CLUSTER-UPDATE send:               2
  CLUSTER-UPDATE recv:               0
  CLUSTER-UPDATE success:            2
  CLUSTER-UPDATE fail:               0
  CLUSTER-UPDATE abort:              0
  CLUSTER-UPDATE timeout:            0
  CLUSTER-UPDATE pending:            0
  CLUSTER-UPDATE max retry reached:  0
  DPD send:                          677
  DPD send fail:                     0
  DPD ACK recv:                      677
  DPD ACK invalid seqno:             0
  IPsec SA policy mismatch:          0
  IPsec SA proposal mismatch:        0
  KEK SA proposal mismatch:          0

Peer gateway: SubSrv02
  Peer IP: 10.16.102.1, Local IP: 10.10.102.1, VR: default
  Role: Sub-server, Status: Active
  CLUSTER-INIT send:                 0
  CLUSTER-INIT recv:                 1
  CLUSTER-INIT success:              1
  CLUSTER-INIT fail:                 0
  CLUSTER-INIT dup:                  0
  CLUSTER-INIT abort:                0
  CLUSTER-INIT timeout:              0
  CLUSTER-UPDATE send:               2
  CLUSTER-UPDATE recv:               0
  CLUSTER-UPDATE success:            2
  CLUSTER-UPDATE fail:               0
  CLUSTER-UPDATE abort:              0
  CLUSTER-UPDATE timeout:            0
  CLUSTER-UPDATE pending:            0
  CLUSTER-UPDATE max retry reached:  0
  DPD send:                          676
  DPD send fail:                     0
  DPD ACK recv:                      676
  DPD ACK invalid seqno:             0
  IPsec SA policy mismatch:          0
  IPsec SA proposal mismatch:        0
  KEK SA proposal mismatch:          0

user@RootSrv> show security group-vpn server statistics
Group: GROUP_ID-0001, Group Id: 1
  Stats:
      Pull Succeeded                : 0
      Pull Failed                   : 0
      Pull Exceed Member Threshold  : 0
      Push Sent                     : 0
      Push Acknowledged             : 0
      Push Unacknowledged           : 0
```

From operational mode, enter the `show security group-vpn server server-cluster`, `show security group-vpn server server-cluster detail`, and `show security group-vpn server statistics` commands on each sub-server.

```
user@SubSrv01> show security group-vpn server server-cluster
Group: GROUP_ID-0001, Group Id: 1
Role: Sub-server, Version Number: 2,
  Peer Gateway                     Peer IP         Role                    Status
  RootSrv                          10.10.101.1     Root-server             Active
```

```
user@SubSrv01> show security group-vpn server server-cluster detail
Group: GROUP_ID-0001, Group Id: 1
Role: Sub-server, Version Number: 2
```

```
Peer gateway: RootSrv
  Peer IP: 10.10.101.1, Local IP: 10.16.101.1, VR: default
  Role: Root-server, Status: Active
  CLUSTER-INIT send:                 1
  CLUSTER-INIT recv:                 0
  CLUSTER-INIT success:              1
  CLUSTER-INIT fail:                 0
  CLUSTER-INIT dup:                  0
  CLUSTER-INIT abort:                0
  CLUSTER-INIT timeout:              0
  CLUSTER-UPDATE send:               0
  CLUSTER-UPDATE recv:               2
  CLUSTER-UPDATE success:            2
  CLUSTER-UPDATE fail:               0
  CLUSTER-UPDATE abort:              0
  CLUSTER-UPDATE timeout:            0
  CLUSTER-UPDATE pending:            0
  CLUSTER-UPDATE max retry reached:  0
  DPD send:                          812
  DPD send fail:                     0
  DPD ACK recv:                      812
  DPD ACK invalid seqno:             0
  IPsec SA policy mismatch:          0
  IPsec SA proposal mismatch:        0
  KEK SA proposal mismatch:          0

user@SubSrv01> show security group-vpn server statistics
Group: GROUP_ID-0001, Group Id: 1
  Stats:
      Pull Succeeded                : 4
      Pull Failed                   : 0
      Pull Exceed Member Threshold  : 0
      Push Sent                     : 8
      Push Acknowledged             : 8
      Push Unacknowledged           : 0
```

Verifying That SAs Are Distributed to Members

Purpose

Verify that the sub-servers have received SAs for distribution to group members and the group members have received the SAs.

Action

From operational mode, enter the `show security group-vpn server kek security-associations` and `show security group-vpn server kek security-associations detail` commands on the root-server.

```
user@RootSrv> show security group-vpn server kek security-associations
Index   Life:sec  Initiator cookie  Responder cookie  GroupId
738885  2888      5742c24020056c6a  d6d479543b56404c  1
```

```
user@RootSrv> show security group-vpn server kek security-associations detail
Index 738885, Group Name: GROUP_ID-0001, Group Id: 1
Initiator cookie: 5742c24020056c6a, Responder cookie: d6d479543b56404c
Authentication method: RSA
Lifetime: Expires in 2883 seconds, Activated
Rekey in 2373 seconds
  Algorithms:
   Sig-hash             : sha256
   Encryption           : aes256-cbc
  Traffic statistics:
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  Server Member Communication: Unicast
  Retransmission Period: 10, Number of Retransmissions: 2
  Group Key Push sequence number: 0

PUSH negotiations in progress: 0
```

From operational mode, enter the `show security group-vpn server kek security-associations` and `show security group-vpn server kek security-associations detail` commands on each sub-server.

```
user@SubSrv01> show security group-vpn server kek security-associations
Index   Life:sec  Initiator cookie  Responder cookie  GroupId
738885  1575      5742c24020056c6a  d6d479543b56404c  1
```

```
user@SubSrv01> show security group-vpn server kek security-associations detail
Index 738879, Group Name: GROUP_ID-0001, Group Id: 1
Initiator cookie: 114e4a214891e42f, Responder cookie: 4b2848d14372e5bd
```

```
Authentication method: RSA
Lifetime: Expires in 4186 seconds, Activated
Rekey in 3614 seconds
  Algorithms:
   Sig-hash              : sha256
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  Server Member Communication: Unicast
  Retransmission Period: 10, Number of Retransmissions: 2
  Group Key Push sequence number: 0

PUSH negotiations in progress: 0
```

From operational mode, enter the `show security group-vpn member kek security-associations` and `show security group-vpn member kek security-associations detail` commands on each group member.

For SRX Series Firewall or vSRX Virtual Firewall group members:

```
user@GM-0001> show security group-vpn server kek security-associations
Index   Server Address  Life:sec  Initiator cookie  Responder cookie  GroupId
5455799 10.17.101.1     1466      5742c24020056c6a  d6d479543b56404c  1

user@GM-0001> show security group-vpn server kek security-associations detail
  Index 5455799, Group Id: 1
  Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.101.1, GDOI Server: 10.17.101.1
  Initiator cookie: 5742c24020056c6a, Responder cookie: d6d479543b56404c
  Lifetime: Expires in 1464 seconds
  Group Key Push Sequence number: 0

  Algorithms:
   Sig-hash              : hmac-sha256-128
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  Stats:
      Push received            :   0
      Delete received          :   0
```

For MX group members:

```
user@GM-0003> show security group-vpn member kek security-associations
Index   Server Address  Life:sec  Initiator cookie  Responder cookie  GroupId
5184329 10.17.101.1     1323      5742c24020056c6a  d6d479543b56404c  1

user@GM-0003> show security group-vpn member kek security-associations detail
  Index 5184329, Group Id: 1
  Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.103.1, GDOI Server: 10.17.101.1
  Initiator cookie: 5742c24020056c6a, Responder cookie: d6d479543b56404c
  Lifetime: Expires in 1321 seconds
  Group Key Push Sequence number: 0

  Algorithms:
   Sig-hash              : hmac-sha256-128
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
  Stats:
      Push received            :   0
      Delete received          :   0
```

Verifying IKE SAs on the Servers

Purpose

Display IKE security associations (SAs) on the servers.

Action

From operational mode, enter the `show security group-vpn server ike security-associations` and `show security group-vpn server ike security-associations detail` commands on the root-server.

```
user@RootSrv> show security group-vpn server ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
738880  UP     2221001e980eb08b  5af00708f5da289c  Main           10.16.104.1
738881  UP     59e8c1d328b1d9fd  d63e823fb8be1f22  Main           10.16.101.1
738883  UP     9cb3a49c6771819e  8df3be8c9ddeb2a7  Main           10.16.102.1
738882  UP     9a8a75f05a1384c5  c6d58696c896b730  Main           10.16.103.1
```

```
user@RootSrv> show security group-vpn server ike security-associations detail
IKE peer 10.16.101.1, Index 738881, Gateway Name: SubSrv01
  Role: Responder, State: UP
  Initiator cookie: 59e8c1d328b1d9fd, Responder cookie: d63e823fb8be1f22
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.10.101.1:848, Remote: 10.16.101.1:848
  Lifetime: Expires in 21890 seconds
  Peer ike-id: 10.16.101.1
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :               150112
   Output bytes  :               153472
   Input  packets:                 1387
   Output packets:                 1387
  Flags: IKE SA is created
IKE peer 10.16.102.1, Index 738883, Gateway Name: SubSrv02
  Role: Responder, State: UP
  Initiator cookie: 9cb3a49c6771819e, Responder cookie: 8df3be8c9ddeb2a7
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.10.102.1:848, Remote: 10.16.102.1:848
  Lifetime: Expires in 21899 seconds
  Peer ike-id: 10.16.102.1
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication       : hmac-sha256-128
   Encryption           : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :               149788
   Output bytes  :               153148
   Input  packets:                 1384
   Output packets:                 1384
  Flags: IKE SA is created
```

From operational mode, enter the `show security group-vpn server ike security-associations` and `show security group-vpn server ike security-associations detail` commands on each sub-server.

```
user@SubSrv01> show security group-vpn server ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
738878  UP     59e8c1d328b1d9fd  d63e823fb8be1f22  Main           10.10.101.1
```

```
user@SubSrv01> show security group-vpn server ike security-associations detail
IKE peer 10.10.101.1, Index 738878, Gateway Name: RootSrv
  Role: Initiator, State: UP
  Initiator cookie: 59e8c1d328b1d9fd, Responder cookie: d63e823fb8be1f22
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.16.101.1:848, Remote: 10.10.101.1:848
  Lifetime: Expires in 20589 seconds
  Peer ike-id: 10.10.101.1
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication       : hmac-sha256-128
   Encryption           : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :               181444
   Output bytes  :               178084
   Input  packets:                 1646
```

```
     Output packets:                 1646
    Flags: IKE SA is created
```

Verifying IPsec SAs on the Servers and Group Members

Purpose

Display IPsec security associations (SAs) on the servers and group members.

Action

From operational mode, enter the `show security group-vpn server ipsec security-associations` and `show security group-vpn server ipsec security-associations detail` commands on the root-server.

```
user@RootSrv> show security group-vpn server ipsec security-associations
Group: GROUP_ID-0001, Group Id: 1
  Total IPsec SAs: 1
  IPsec SA          Algorithm        SPI               Lifetime
  GROUP_ID-0001     ESP:aes-256/sha256 dddef414        2773

user@RootSrv> show security group-vpn server ipsec security-associations detail
Group: GROUP_ID-0001, Group Id: 1
Total IPsec SAs: 1
  IPsec SA: GROUP_ID-0001
    Protocol: ESP, Authentication: sha256, Encryption: aes-256
    Anti-replay: D3P enabled
    SPI: dddef414
    Lifetime: Expires in 1670 seconds, Activated
    Rekey in 1160 seconds
    Policy Name: 1
      Source: 172.16.0.0/12
      Destination: 172.16.0.0/12
      Source Port: 0
      Destination Port: 0
      Protocol: 0
```

From operational mode, enter the show security group-vpn server ipsec security-associations and show security group-vpn server ipsec security-associations detail commands on each sub-server.

```
user@SubSrv01> show security group-vpn server ipsec security-associations
Group: GROUP_ID-0001, Group Id: 1
  Total IPsec SAs: 1
  IPsec SA          Algorithm        SPI              Lifetime
  GROUP_ID-0001     ESP:aes-256/sha256 dddef414       1520

user@SubSrv01> show security group-vpn server ipsec security-associations detail
Group: GROUP_ID-0001, Group Id: 1
Total IPsec SAs: 1
  IPsec SA: GROUP_ID-0001
    Protocol: ESP, Authentication: sha256, Encryption: aes-256
    Anti-replay: D3P enabled
    SPI: dddef414
    Lifetime: Expires in 1518 seconds, Activated
    Rekey in 1230 seconds
    Policy Name: 1
      Source: 172.16.0.0/12
      Destination: 172.16.0.0/12
      Source Port: 0
      Destination Port: 0
      Protocol: 0
```

From operational mode, enter the show security group-vpn member ipsec security-associations and show security group-vpn member ipsec security-associations detail commands on each group member

For SRX Series Firewall or vSRX Virtual Firewall group members:

```
user@GM-0001> show security group-vpn member ipsec security-associations
  Total active tunnels: 1
  ID    Server           Port  Algorithm       SPI      Life:sec/kb  GId lsys
  <>49152 10.17.101.1    848   ESP:aes-256/sha256-128 dddef414 1412/ unlim 1 root

user@GM-0001> show security group-vpn member ipsec security-associations detail
  Virtual-system: root Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.101.1, GDOI Server: 10.17.101.1
  Group Id: 1
  Routing Instance: default
  Recovery Probe: Enabled
  DF-bit: clear
  Stats:
      Pull Succeeded             :   1
      Pull Failed                :   0
      Pull Timeout               :   0
      Pull Aborted               :   0
      Push Succeeded             :   2
      Push Failed                :   0
      Server Failover            :   0
      Delete Received            :   0
      Exceed Maximum Keys(4)     :   0
      Exceed Maximum Policies(10):   0
      Unsupported Algo           :   0
  Flags:
      Rekey Needed:   no

    List of policies received from server:
    Tunnel-id: 49152
      Source IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)
      Destination IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)

      Direction: bi-directional, SPI: dddef414
      Protocol: ESP, Authentication: sha256-128, Encryption: aes-256
      Hard lifetime: Expires in 1409 seconds, Activated
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expires in 1193 seconds
      Mode: Tunnel, Type: Group VPN, State: installed
      Anti-replay service: D3P enabled
```

For MX group members:

```
user@GM-0003> show security group-vpn member ipsec security-associations
  Total active tunnels: 1
  ID    Server           Port  Algorithm       SPI      Life:sec/kb  GId lsys
  <>10001 10.17.101.1    848   ESP:aes-256/sha256-128 dddef414 1308/ unlim 1 root

user@GM-0003> show security group-vpn member ipsec security-associations detail
  Virtual-system: root Group VPN Name: GROUP_ID-0001
  Local Gateway: 10.18.103.1, GDOI Server: 10.17.101.1
  Group Id: 1
  Rule Match Direction: output,  Tunnel-MTU: 1400
  Routing Instance: default
  DF-bit: clear
```

```
Stats:
    Pull Succeeded            :   1
    Pull Failed               :   0
    Pull Timeout              :   0
    Pull Aborted              :   0
    Push Succeeded            :   2
    Push Failed               :   0
    Server Failover           :   0
    Delete Received           :   0
    Exceed Maximum Keys(4)    :   0
    Exceed Maximum Policies(1):   0
    Unsupported Algo          :   0
Flags:
    Rekey Needed:   no

  List of policies received from server:
  Tunnel-id: 10001
    Source IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)
    Destination IP: ipv4_subnet(any:0,[0..7]=172.16.0.0/12)

    Direction: bi-directional, SPI: dddef414
    Protocol: ESP, Authentication: sha256-128, Encryption: aes-256
    Hard lifetime: Expires in 1305 seconds, Activated
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1087 seconds
    Mode: Tunnel, Type: Group VPN, State: installed
    Anti-replay service: D3P enabled
```

Verifying IPsec Policies on Group Members

Purpose

Display the IPsec policy on an SRX Series Firewall or vSRX Virtual Firewall group member.

This command is not available for MX Series group members.

Action

From operational mode, enter the **show security group-vpn member policy** command on SRX Series Firewall or vSRX Virtual Firewall group members.

```
user@GM-0001> show security group-vpn member policy
Group VPN Name: GROUP_ID-0001, Group Id: 1
From-zone: LAN, To-zone: WAN
  Tunnel-id: 49152, Policy type: Secure
    Source      : IP <172.16.0.0 - 172.31.255.255>, Port <0 - 65535>, Protocol <0>
    Destination : IP <172.16.0.0 - 172.31.255.255>, Port <0 - 65535>, Protocol <0>

  Tunnel-id: 63488, Policy type: Fail-close
    Source      : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <0>
    Destination : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <0>
```

SEE ALSO

Group VPNv2 Configuration Overview | 763

Configuring Group VPNs in Group VPNv2 on Routing Device

RELATED DOCUMENTATION

Group VPNv1 | 703

11

CHAPTER

ADVPN

Auto Discovery VPNs

IN THIS SECTION

Auto Discovery VPN (ADVPN) dynamically establishes VPN tunnels between spokes to avoid routing traffic through the Hub.

Understanding Auto Discovery VPN

IN THIS SECTION

Auto Discovery VPN (ADVPN) is a technology that allows the central HUB to dynamically inform spokes about a better path for traffic between two spokes. When both spokes acknowledge the information from the HUB, they establish a shortcut tunnel and change the routing topology for the host to reach the other side without sending traffic through the HUB.

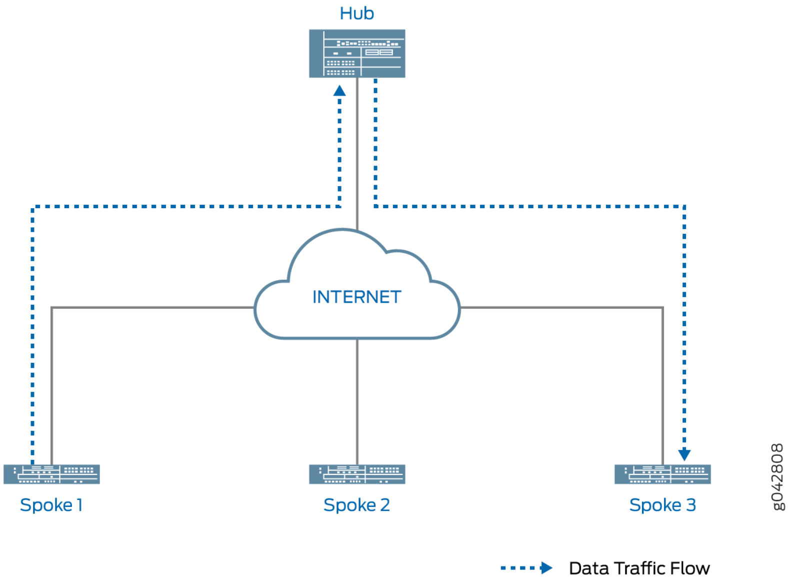
ADVPN Protocol

ADVPN use an extension of IKEv2 protocol to exchange messages between two peers, which allows the spokes to establish a shortcut tunnel between each other. Devices that support the ADVPN extension send an `ADVPN_SUPPORTED` notification in the IKEv2 Notify payload including its capability information and the ADVPN version number during the initial IKE exchange. A device that supports ADVPN can act as either a *shortcut suggester* or a shortcut partner, but not both.

Establishing a Shortcut

An IPsec VPN gateway can act as a *shortcut suggester* when it notices that traffic is exiting a tunnel with one of its peers and entering a tunnel with another peer. Figure 58 on page 920 shows traffic from Spoke 1 to Spoke 3 passing through the hub.

Figure 58: Spoke-to-Spoke Traffic Passing Through Hub

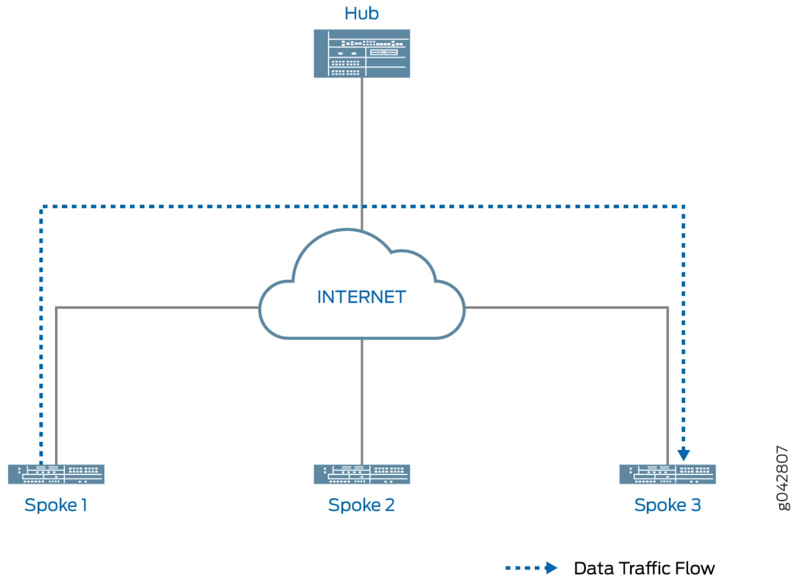


When ADVPN is configured on the devices, ADVPN shortcut capability information is exchanged between the hub and the spokes. As long as Spokes 1 and 3 have previously advertised ADVPN shortcut partner capability to the hub, the hub can suggest that Spokes 1 and 3 establish a shortcut between each other.

The shortcut suggester uses its already established IKEv2 SAs with the peers to begin a shortcut exchange with one of the two peers. If the peer accepts the shortcut exchange, then the shortcut suggester begins a shortcut exchange with the other peer. The shortcut exchange includes information to allow the peers (referred to as *shortcut partners*) to establish IKE and IPsec SAs with each other. The creation of the shortcut between the shortcut partners starts only after both peers accept the shortcut exchange.

Figure 59 on page 921 shows traffic passing through a shortcut between Spokes 1 and 3. Traffic from Spoke 1 to Spoke 3 does not need to traverse the hub.

Figure 59: Spoke-to-Spoke Traffic Passing Through Shortcut



Shortcut Initiator and Responder Roles

The shortcut suggester chooses one of the shortcut partners to act as the initiator for the shortcut; the other partner acts as the responder. If one of the partners is behind a NAT device, then the partner behind the NAT device is chosen as the initiator. If none of the partners is behind a NAT device, then the suggester randomly chooses one of the partners as the initiator; the other partner acts as the responder. If both partners are behind NAT devices, then a shortcut cannot be created between them; the suggester does not send a shortcut exchange to any of the peers.

The shortcut suggester begins the shortcut exchange with the responder first. If the responder accepts the shortcut suggestion, then the suggester notifies the initiator.

Using information contained in the shortcut suggester’s notification, the shortcut initiator establishes an IKEv2 exchange with the responder, and a new IPsec SA is established between the two partners. On each partner, the route to the network behind its partner now points to the shortcut instead of to the tunnel between the partner and the suggester. Traffic originating behind one of the partners that is destined to a network behind the other shortcut partner flows over the shortcut.

If the partners decline the shortcut suggestion, then the partners notify the suggester with the reason for the rejection. In this case, traffic between the partners continues to flow through the shortcut suggester.

Shortcut Attributes

The shortcut receives some of its attributes from the shortcut suggester while other attributes are inherited from the suggester-partner VPN tunnel configuration. Table 83 on page 922 shows the parameters of the shortcut.

Table 83: Shortcut Parameters

Attributes	Received/Inherited From
ADVPN	Configuration
Antireplay	Configuration
Authentication algorithm	Configuration
Dead peer detection	Configuration
DF bit	Configuration
Encryption algorithm	Configuration
Establish tunnels	Suggester
External interface	Configuration
Gateway policy	Configuration
General IKE ID	Configuration
IKE version	Configuration

Table 83: Shortcut Parameters *(Continued)*

Attributes	Received/Inherited From
Install interval	Configuration
Local address	Configuration
Local identity	Suggester
NAT traversal	Configuration
Perfect forward secrecy	Configuration
Protocol	Configuration
Proxy ID	Not applicable
Remote address	Suggester
Remote identity	Suggester
Respond bad SPI	Configuration
Traffic selector	Not applicable

Shortcut Termination

By default, the shortcut lasts indefinitely. Shortcut partners terminate the shortcut if traffic falls below a specified rate for a specified time. By default, the shortcut is terminated if traffic falls below 5 packets per second for 300 seconds; the idle time and idle threshold values are configurable for partners. The shortcut can be manually deleted on either shortcut partner with the `clear security ike security-association` or `clear security ipsec security-association` commands to clear the corresponding IKE or IPsec SA. Either of the shortcut partners can terminate the shortcut at any time by sending an IKEv2 delete payload to the other shortcut partner.

When the shortcut is terminated, the corresponding IKE SA and all child IPsec SAs are deleted. After the shortcut is terminated, the corresponding route is deleted on both shortcut partners and traffic between the two peers again flows through the suggester. Shortcut termination information is sent from a partner to the suggester.

The lifetime of a shortcut is independent of the tunnel between the shortcut suggester and shortcut partner. The shortcut is not terminated simply because the tunnel between the suggester and partner is terminated.

ADVPN Configuration Limitations

Note the following limitations when configuring ADVPN:

- ADVPN is only supported for site-to-site communications. Configuring an ADVPN suggester is only allowed on AutoVPN hubs.
- You cannot configure both suggester and partner roles. When ADVPN is enabled on a gateway, you cannot disable both suggester and partner roles on the gateway.
- As mentioned previously, you cannot create a shortcut between partners that are both behind NAT devices. The suggester can initiate a shortcut exchange if only one of the partners is behind a NAT device or if no partners are behind NAT devices.
- Multicast traffic is not supported.

  1. Starting in Junos OS Release 19.2R1, on SRX300, SRX320, SRX340, SRX345, SRX550, SRX1500, vSRX Virtual Firewall 2.0 (with 2 vCPUs), and vSRX Virtual Firewall 3.0 (with 2 vCPUs) Series devices, Protocol Independent Multicast (PIM) using point-to-multipoint (P2MP) mode supports Auto Discovery VPN in which a new `p2mp` interface type is introduced for PIM. The `p2mp` interface tracks all PIM joins per neighbor to ensure multicast forwarding or replication only happens to those neighbors that are in joined state.
  2. Starting with Junos OS Release 18.1R1, ADVPN supports IPv6.

The following configurations are not supported with ADVPN:

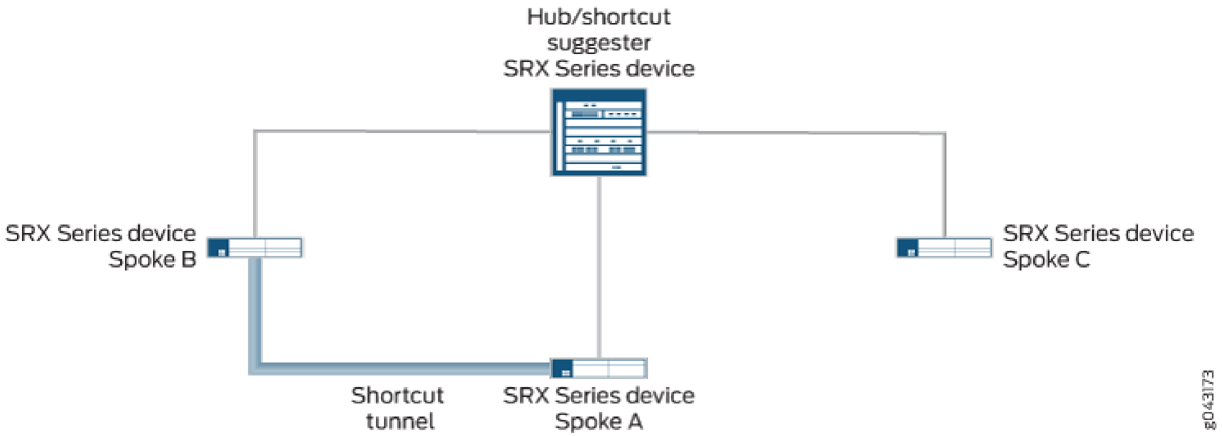
- IKEv1
- Policy-based VPN
- IKEv2 configuration payload
- Traffic selectors
- Preshared key
- Point-to-point secure tunnel interfaces

Understanding Traffic Routing with Shortcut Tunnels

Tunnel flaps or catastrophic changes can cause both static tunnels and shortcut tunnels to go down. When this happens, traffic to a specific destination might be routed through an unexpected shortcut tunnel instead of through an expected static tunnel.

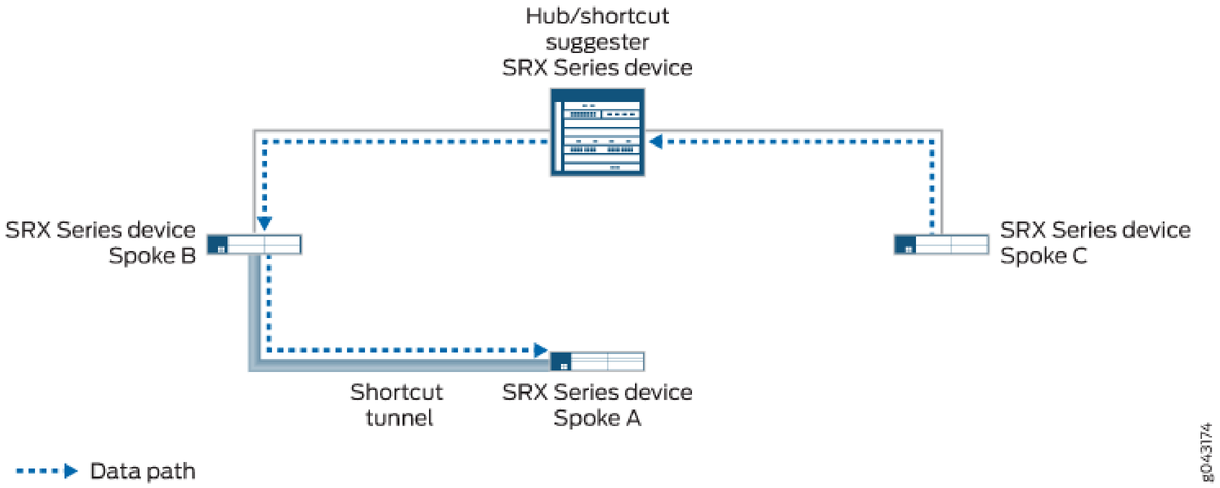
In Figure 60 on page 925, static tunnels exist between the hub and each of the spokes. OSPF adjacencies are established between the hub and spokes. Spoke A also has a shortcut tunnel with Spoke B and OSPF adjacencies are established between the spokes. The hub (the shortcut suggester) recognizes that if connectivity between the hub and Spoke A goes down, Spoke A's network can be reached through the shortcut tunnel between Spoke B and Spoke A.

Figure 60: Static Tunnels and Shortcut Tunnel Established in Hub-and-Spoke Network


Hub/shortcut suggester SRX Series device
SRX Series device Spoke B
SRX Series device Spoke C
Shortcut tunnel
SRX Series device Spoke A
g043173

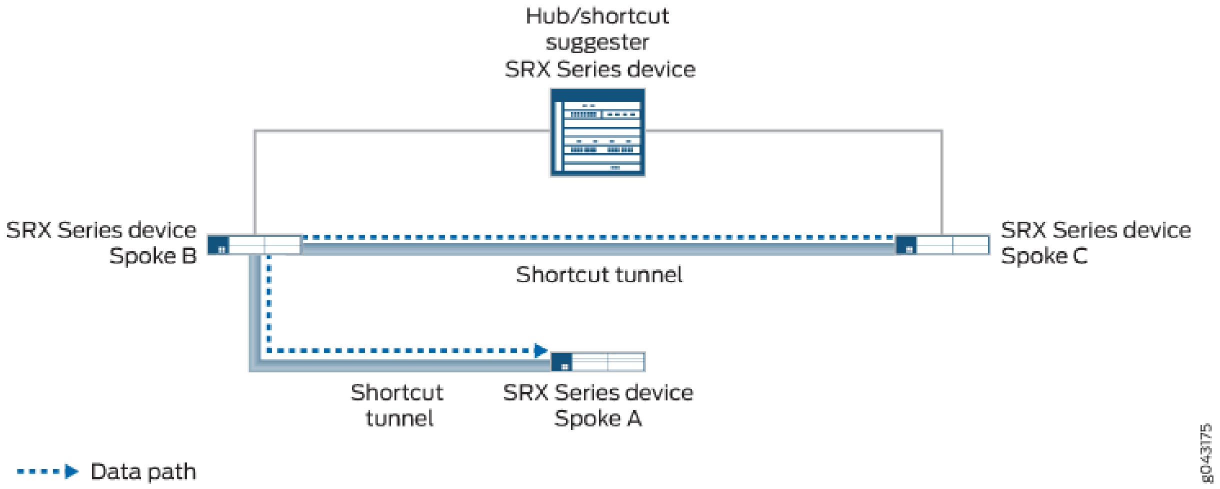
In Figure 61 on page 926, the static tunnel between the hub and Spoke A is down. If there is new traffic from Spoke C to Spoke A, Spoke C forwards the traffic to the hub because it does not have a shortcut tunnel with Spoke A. The hub does not have an active static tunnel with Spoke A but it recognizes that there is a shortcut tunnel between Spoke A and Spoke B, so it forwards the traffic from Spoke C to Spoke B.

Figure 61: Traffic Path from Spoke C to Spoke A



As long as both Spoke B and Spoke C support Auto Discovery VPN (ADVPN) partner capability, the hub can suggest that the spokes establish a direct shortcut between each other. This occurs even though there is no direct traffic between the two spokes. Traffic from Spoke C to Spoke A travels through the shortcut tunnel between Spoke C and Spoke B, and then through the shortcut tunnel between Spoke B and Spoke A (see Figure 62 on page 926).

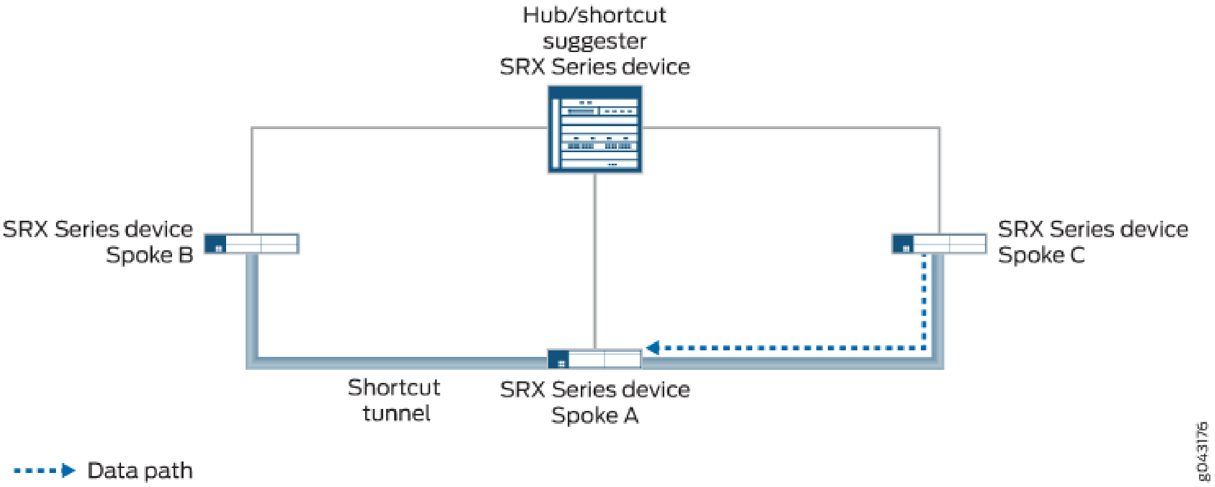
Figure 62: Traffic Path from Spoke C to Spoke A Through Shortcut Tunnels



When the static tunnel between the hub and Spoke A is reestablished, the tunnel is advertised to all spokes. Spoke C learns that there is a better route to reach Spoke A; instead of passing traffic through

Spoke B, it forwards traffic for Spoke A to the hub. The hub suggests that a shortcut tunnel be established between Spoke C and Spoke A. When the shortcut tunnel is established between Spoke C and Spoke A, traffic flows through the shortcut tunnel (see Figure 63 on page 927). Traffic between Spoke C and Spoke A no longer travels through Spoke B, and the shortcut tunnel between Spoke B and Spoke C eventually disappears.

Figure 63: Traffic Path from Spoke C to Spoke A Through Shortcut Tunnel



You can use the `connection-limit` option at the `[edit security ike gateway` *`gateway-name`* `advpn partner]` hierarchy level to set the maximum number of shortcut tunnels that can be created with different shortcut partners using a particular gateway. The maximum number, which is also the default, is platform-dependent.

SEE ALSO

Understanding Hub-and-Spoke VPNs | 187

Example: Improving Network Resource Utilization with Auto Discovery VPN Dynamic Tunnels

IN THIS SECTION

If you are deploying an AutoVPN network, you might be able to increase your network resource utilization by configuring Auto Discovery VPN (ADVPN). In AutoVPN networks, VPN traffic flows through the hub even when the traffic is travelling from one spoke to another. ADVPN allows VPN tunnels to be established dynamically between spokes, which can result in better network resource utilization. Use this example to configure ADVPN to enable dynamic spoke-to-spoke VPN tunnels in your AutoVPN network.

Requirements

This example uses the following hardware and software components:

- Three supported SRX Series Firewalls as AutoVPN hub and spokes.

- Junos OS Release 12.3X48-D10 or later releases that support ADVPN.

- Digital certificates enrolled in the hub and spokes that allow the devices to authenticate each other.

Before you begin:

1. Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates. See "Understanding Local Certificate Requests" on page 52.

2. Enroll the digital certificates in each device. See "Example: Loading CA and Local Certificates Manually" on page 63.

This example uses the OSPF dynamic routing protocol as well as static route configurations to forward packets through VPN tunnels. You should be familiar with the OSPF dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

This example shows the configurations of an AutoVPN hub and two spokes for ADVPN. The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as to access resources on the hub. While traffic is initially passed from one spoke to the other through the hub, ADVPN allows the spokes to establish a direct security association between each other. The hub acts as the shortcut suggester. On the hub, the ADVPN configuration disables the `partner` role. On the spokes, ADVPN configuration disables the `suggester` role.

Certain Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and spokes must have the same values. Table 84 on page 929 shows the values used in this example.

Table 84: Phase 1 and Phase 2 Options for AutoVPN Hub and Spokes for ADVPN Example

Option	Value
IKE proposal:	
Authentication method	rsa-signatures
Diffie-Hellman (DH) group	group5
Authentication algorithm	sha1
Encryption algorithm	aes-256-cbc
IKE policy:	
Certificate	local-certificate
IKE gateway:	

Table 84: Phase 1 and Phase 2 Options for AutoVPN Hub and Spokes for ADVPN Example *(Continued)*

Option	Value
Version	v2-only
IPsec proposal:	
Protocol	esp
Authentication algorithm	hmac-sha1-96
Encryption algorithm	aes-256-cbc
IPsec policy:	
Perfect Forward Secrecy (PFS) group	group5

The IKE gateway configuration on the hub and spokes include remote and local values that identify VPN peers. Table 85 on page 930 shows the IKE gateway configuration for the hub and spokes in this example.

Table 85: IKE Gateway Configuration for ADVPN Example

Option	Hub	Spokes
Remote IP address	Dynamic	Spoke 1: 11.1.1.1 Spoke 2: 11.1.1.1
Local IP address	11.1.1.1	Spoke 1: 21.1.1.2 Spoke 2: 31.1.1.2
Remote IKE ID	Distinguished name (DN) with the string “XYZ” in the organization (O) field and “Sales” in the organization unit (OU) field in the spokes’ certificates	DN with the string “Sales” in the OU field in the hub’s certificate

Table 85: IKE Gateway Configuration for ADVPN Example *(Continued)*

Option	Hub	Spokes
Local IKE ID	DN on the hub's certificate	DN on the spokes' certificate

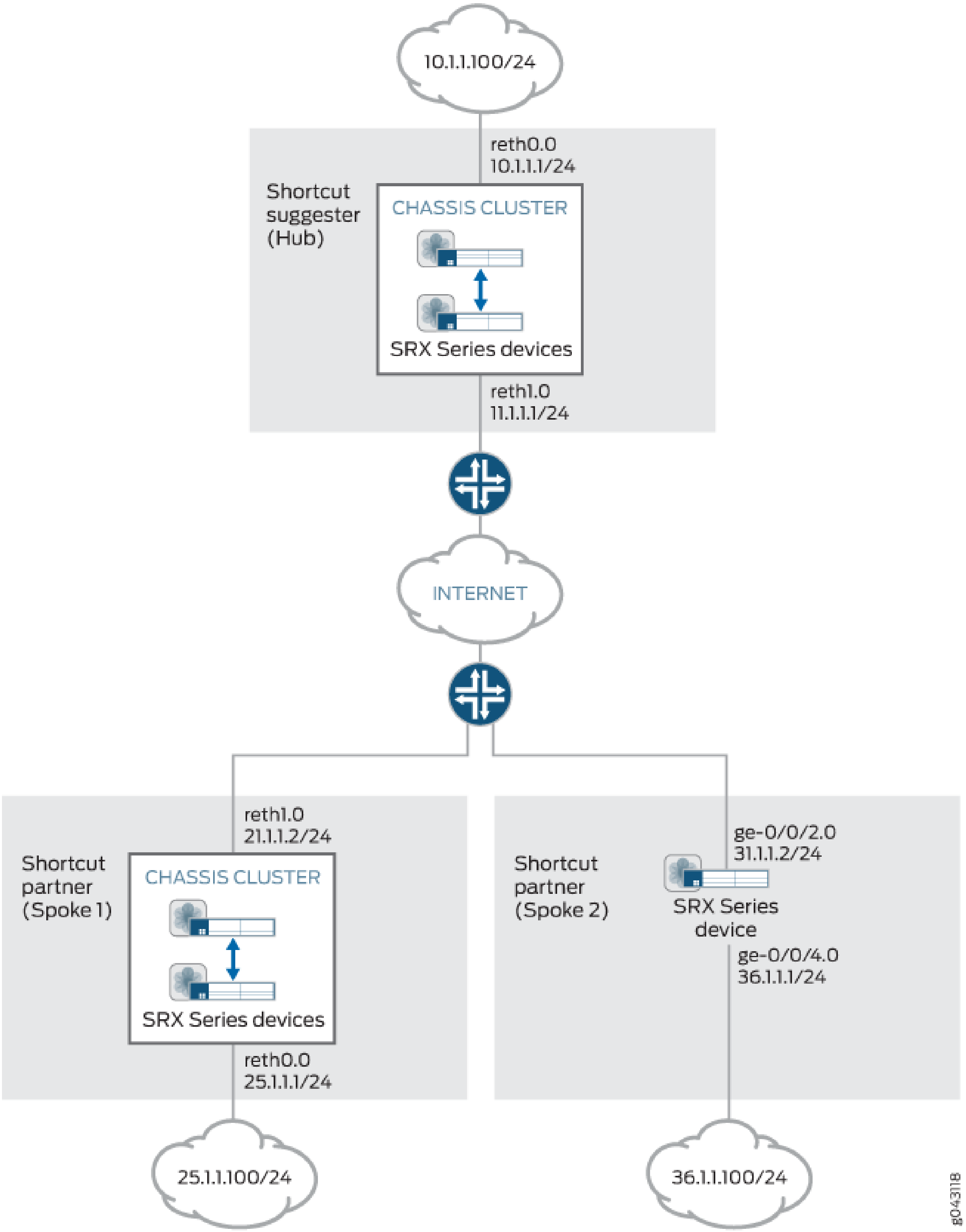
The hub authenticates the spokes' IKE ID if the subject fields of the spokes' certificates contain the string "XYZ" in the O field and "Sales" in the OU field.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Topology

Figure 64 on page 932 shows the SRX Series Firewalls to be configured for this example.

Figure 64: AutoVPN Deployment with ADVPN



Configuration

IN THIS SECTION

Configuring the Suggester (Hub)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/3 gigether-options redundant-parent reth0
set interfaces ge-0/0/4 gigether-options redundant-parent reth1
set interfaces ge-7/0/3 gigether-options redundant-parent reth0
set interfaces ge-7/0/4 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.1.1.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 11.1.1.1/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 172.16.1.1/24
set protocols ospf graceful-restart restart-duration 300
set protocols ospf graceful-restart notify-duration 300
set protocols ospf graceful-restart no-strict-lsa-checking
set protocols ospf area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.1 metric 10
set protocols ospf area 0.0.0.0 interface st0.1 retransmit-interval 1
set protocols ospf area 0.0.0.0 interface st0.1 dead-interval 40
set protocols ospf area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf area 0.0.0.0 interface reth0.0
set routing-options graceful-restart
set routing-options static route 21.1.1.0/24 next-hop 11.1.1.2
set routing-options static route 31.1.1.0/24 next-hop 11.1.1.2
set routing-options router-id 172.16.1.1
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate Suggester_Certificate_ID
set security ike gateway SUGGESTER_GW ike-policy IKE_POL
set security ike gateway SUGGESTER_GW dynamic distinguished-name wildcard O=XYZ, OU=Sales
set security ike gateway SUGGESTER_GW dynamic ike-user-type group-ike-id
set security ike gateway SUGGESTER_GW dead-peer-detection
set security ike gateway SUGGESTER_GW local-identity distinguished-name
set security ike gateway SUGGESTER_GW external-interface reth1.0
set security ike gateway SUGGESTER_GW local-address 11.1.1.1
set security ike gateway SUGGESTER_GW advpn partner disable
set security ike gateway SUGGESTER_GW advpn suggester
set security ike gateway SUGGESTER_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn SUGGESTER_VPN bind-interface st0.1
set security ipsec vpn SUGGESTER_VPN ike gateway SUGGESTER_GW
set security ipsec vpn SUGGESTER_VPN ike ipsec-policy IPSEC_POL
set security pki ca-profile advpn ca-identity advpn
set security pki ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces reth0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces reth1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the suggester:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/3 gigether-options redundant-parent reth0
user@host# set ge-0/0/4 gigether-options redundant-parent reth1
user@host# set ge-7/0/3 gigether-options redundant-parent reth0
user@host# set ge-7/0/4 gigether-options redundant-parent reth1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet address 10.1.1.1/24
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 11.1.1.1/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 172.16.1.1/24
```

2. Configure the routing protocol and static routes.

```
[edit protocols ospf]
user@host# set graceful-restart restart-duration 300
user@host# set graceful-restart notify-duration 300
user@host# set graceful-restart no-strict-lsa-checking
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 metric 10
user@host# set area 0.0.0.0 interface st0.1 retransmit-interval 1
user@host# set area 0.0.0.0 interface st0.1 dead-interval 40
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface reth0.0
[edit routing-options]
user@host# set graceful-restart
user@host# set static route 21.1.1.0/24 next-hop 11.1.1.2
user@host# set static route 31.1.1.0/24 next-hop 11.1.1.2
user@host# set router-id 172.16.1.1
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
```

```
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate Suggester_Certificate_ID
[edit security ike gateway SUGGESTER_GW]
user@host# set ike-policy IKE_POL
user@host# set dynamic distinguished-name wildcard O=XYZ, OU=Sales
user@host# set dynamic ike-user-type group-ike-id
user@host# set dead-peer-detection
user@host# set local-identity distinguished-name
user@host# set external-interface reth1.0
user@host# set local-address 11.1.1.1
user@host# set advpn partner disable
user@host# set advpn suggester
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals IPSEC_PROP
[edit security isec vpn SUGGESTER_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway SUGGESTER_GW
user@host# set ike ipsec-policy IPSEC_POL
```

5. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile advpn ca-identity advpn
user@host# set ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
```

6. Configure zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces reth0.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces reth1.0
```

7. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
    user@host# show interfaces
    ge-0/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    ge-0/0/4 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-7/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    ge-7/0/4 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 10.1.1.1/24;
            }
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 11.1.1.1/24;
            }
        }
    }
    st0 {
        unit 1 {
            multipoint;
            family inet {
                address 172.16.1.1/24;
            }
        }
    }
    [edit]
    user@host# show protocols
    ospf {
        graceful-restart {
            restart-duration 300;
            notify-duration 300;
            no-strict-lsa-checking;
        }
        area 0.0.0.0 {
            interface st0.1 {
                interface-type p2mp;
                metric 10;
                retransmit-interval 1;
                dead-interval 40;
                demand-circuit;
                dynamic-neighbors;
            }
            interface reth0.0;
        }
    }
    [edit]
    user@host# show routing-options
    graceful-restart;
    static {
        route 21.1.1.0/24 next-hop 11.1.1.2;
        route 31.1.1.0/24 next-hop 11.1.1.2;
    }
    router-id 172.16.1.1;
    [edit]
    user@host# show security ike
    proposal IKE_PROP {
        authentication-method rsa-signatures;
        dh-group group5;
        authentication-algorithm sha1;
        encryption-algorithm aes-256-cbc;
    }
    policy IKE_POL {
        proposals IKE_PROP;
        certificate {
            local-certificate Suggester_Certificate_ID;
        }
    }
    gateway SUGGESTER_GW {
        ike-policy IKE_POL;
        dynamic {
            distinguished-name {
                wildcard O=XYZ, OU=Sales;
            }
            ike-user-type group-ike-id;
        }
        dead-peer-detection {
        }
        local-identity distinguished-name;
        external-interface reth1.0
        local-address 11.1.1.1;
        advpn {
            partner {
                disable;
                }
                suggester {
            ]
        }
        version v2-only;
    }
    [edit]
    user@host# show security ipsec
    proposal IPSEC_PROP {
        protocol esp;
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm aes-256-cbc;
    }
    policy IPSEC_POL {
        perfect-forward-secrecy {
            keys group5;
        }
        proposals IPSEC_PROP;
    }
    vpn SUGGESTER_VPN {
        bind-interface st0.1;
        ike {
            gateway SUGGESTER_GW;
            ipsec-policy IPSEC_POL;
        }
    }
    [edit]
    user@host# show security pki
    ca-profile advpn {
        ca-identity advpn;
        enrollment {
            url http://10.157.92.176:8080/scep/advpn/;
        }
    }
    [edit]
    user@host# show security zones
```

```
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            st0.1;
            reth0.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            reth1.0;
        }
    }
    [edit]
user@host# show security policies
    default-policy {
        permit-all;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Partner (Spoke 1)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/3 gigether-options redundant-parent reth0
set interfaces ge-0/0/4 gigether-options redundant-parent reth1
set interfaces ge-7/0/3 gigether-options redundant-parent reth0
set interfaces ge-7/0/4 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 25.1.1.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 21.1.1.2/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 172.16.1.2/24
set protocols ospf graceful-restart restart-duration 300
set protocols ospf graceful-restart notify-duration 300
set protocols ospf graceful-restart no-strict-lsa-checking
set protocols ospf area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.1 metric 15
set protocols ospf area 0.0.0.0 interface st0.1 retransmit-interval 1
set protocols ospf area 0.0.0.0 interface st0.1 dead-interval 40
set protocols ospf area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf area 0.0.0.0 interface reth0.0
set routing-options graceful-restart
set routing-options static route 11.1.1.0/24 next-hop 21.1.1.1
set routing-options static route 31.1.1.0/24 next-hop 21.1.1.1
set routing-options router-id 172.16.1.2
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate Partner1_Certificate_ID
set security ike gateway PARTNER_GW ike-policy IKE_POL
set security ike gateway PARTNER_GW address 11.1.1.1
set security ike gateway PARTNER_GW local-identity distinguished-name
set security ike gateway PARTNER_GW remote-identity distinguished-name container OU=Sales
set security ike gateway PARTNER_GW external-interface reth1
set security ike gateway PARTNER_GW local-address 21.1.1.2
set security ike gateway PARTNER_GW advpn suggester disable
set security ike gateway PARTNER_GW advpn partner
set security ike gateway PARTNER_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn PARTNER_VPN bind-interface st0.1
set security ipsec vpn PARTNER_VPN ike gateway PARTNER_GW
set security ipsec vpn PARTNER_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn PARTNER_VPN establish-tunnels immediately
set security pki ca-profile advpn ca-identity advpn
set security pki ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces reth0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces reth1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/3 gigether-options redundant-parent reth0
user@host# set ge-0/0/4 gigether-options redundant-parent reth1
user@host# set ge-7/0/3 gigether-options redundant-parent reth0
user@host# set ge-7/0/4 gigether-options redundant-parent reth1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet address 25.1.1.1/24
user@host# set reth1 redundant-ether-options redundancy-group 1
```

```
user@host# set reth1 unit 0 family inet address 21.1.1.2/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 172.16.1.2/24
```

2. Configure the routing protocol and static routes.

```
[edit protocols ospf]
user@host# set graceful-restart restart-duration 300
user@host# set graceful-restart notify-duration 300
user@host# set graceful-restart no-strict-lsa-checking
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 metric 15
user@host# set area 0.0.0.0 interface st0.1 retransmit-interval 1
user@host# set area 0.0.0.0 interface st0.1 dead-interval 40
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set protocols ospf area 0.0.0.0 interface reth0.0
[edit routing-options]
user@host# set graceful-restart
user@host# set static route 11.1.1.0/24 next-hop 21.1.1.1
user@host# set static route 31.1.1.0/24 next-hop 21.1.1.1
user@host# set router-id 172.16.1.2
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate Partner1_Certificate_ID
[edit security ike gateway PARTNER_GW]
user@host# set ike-policy IKE_POL
user@host# set address 11.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=Sales
user@host# set external-interface reth1
user@host# set local-address 21.1.1.2
user@host# set advpn suggester disable
```

```
user@host# set advpn partner
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals IPSEC_PROP
[edit security isec vpn PARTNER_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway PARTNER_GW
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

5. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile advpn ca-identity advpn
user@host# set ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
```

6. Configure zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces reth0.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces reth1.0
```

7. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
    user@host# show interfaces
    ge-0/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    ge-0/0/4 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-7/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    ge-7/0/4 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
```

```
            address 25.1.1.1/24;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 21.1.1.2/24;
        }
    }
}
st0 {
    unit 1 {
        multipoint;
        family inet {
            address 172.16.1.2/24;
        }
    }
}
[edit]
user@host# show protocols
ospf {
    graceful-restart {
        restart-duration 300;
        notify-duration 300;
        no-strict-lsa-checking;
    }
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            metric 15;
            retransmit-interval 1;
            dead-interval 40;
            demand-circuit;
            dynamic-neighbors;
        }
        interface reth0.0;
    }
}
[edit]
```

```
user@host# show routing-options
graceful-restart;
static {
    route 11.1.1.0/24 next-hop 21.1.1.1;
    route 31.1.1.0/24 next-hop 21.1.1.1;
}
router-id 172.16.1.2;
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    certificate {
        local-certificate Partner1_Certificate_ID;
    }
}
gateway PARTNER_GW {
    ike-policy IKE_POL;
    address 11.1.1.1;
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=Sales;
    external-interface reth1;
    local-address 21.1.1.2;
    advpn {
        suggester {
            disable;
        }
        partner {
        }
    }
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-256-cbc;
```

```
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals IPSEC_PROP;
}
vpn PARTNER_VPN {
    bind-interface st0.1;
    ike {
        gateway PARTNER_GW;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security pki
ca-profile advpn {
    ca-identity advpn;
    enrollment {
        url http://10.157.92.176:8080/scep/advpn/;
    }
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.1;
        reth0.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            reth1.0;
        }
    }
    [edit]
user@host# show security policies
    default-policy {
        permit-all;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Partner (Spoke 2)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/2 unit 0 family inet address 31.1.1.2/24
set interfaces ge-0/0/4 unit 0 family inet address 36.1.1.1/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 172.16.1.3/24
set protocols ospf graceful-restart restart-duration 300
set protocols ospf graceful-restart notify-duration 300
set protocols ospf graceful-restart no-strict-lsa-checking
set protocols ospf area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.1 metric 15
set protocols ospf area 0.0.0.0 interface st0.1 retransmit-interval 1
set protocols ospf area 0.0.0.0 interface st0.1 dead-interval 40
set protocols ospf area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf area 0.0.0.0 interface ge-0/0/4.0
set routing-options graceful-restart
set routing-options static route 11.1.1.0/24 next-hop 31.1.1.1
set routing-options static route 21.1.1.0/24 next-hop 31.1.1.1
set routing-options router-id 172.16.1.3
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group5
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate Partner2_Certificate_ID
set security ike gateway PARTNER_GW ike-policy IKE_POL
set security ike gateway PARTNER_GW address 11.1.1.1
set security ike gateway PARTNER_GW dead-peer-detection
set security ike gateway PARTNER_GW local-identity distinguished-name
set security ike gateway PARTNER_GW remote-identity distinguished-name container OU=Sales
set security ike gateway PARTNER_GW external-interface ge-0/0/2.0
set security ike gateway PARTNER_GW local-address 31.1.1.2
set security ike gateway PARTNER_GW advpn suggester disable
set security ike gateway PARTNER_GW advpn partner
set security ike gateway PARTNER_GW version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group5
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn PARTNER_VPN bind-interface st0.1
set security ipsec vpn PARTNER_VPN ike gateway PARTNER_GW
set security ipsec vpn PARTNER_VPN ike ipsec-policy IPSEC_POL
set security ipsec vpn PARTNER_VPN establish-tunnels immediately
set security pki ca-profile advpn ca-identity advpn
set security pki ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/4.0
set security zones security-zone trust interfaces st0.1
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/2.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/2 unit 0 family inet address 31.1.1.2/24
user@host# set ge-0/0/4 unit 0 family inet address 36.1.1.1/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 172.16.1.3/24
```

2. Configure the routing protocol and static routes.

```
[edit protocols ospf
user@host# set graceful-restart restart-duration 300
user@host# set graceful-restart notify-duration 300
user@host# set graceful-restart no-strict-lsa-checking
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 metric 15
user@host# set area 0.0.0.0 interface st0.1 retransmit-interval 1
user@host# set area 0.0.0.0 interface st0.1 dead-interval 40
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/4.0
[edit routing-options]
user@host# set graceful-restart
user@host# set static route 11.1.1.0/24 next-hop 31.1.1.1
user@host# set static route 21.1.1.0/24 next-hop 31.1.1.1
user@host# set router-id 172.16.1.3
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy IKE_POL]
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate Partner2_Certificate_ID
[edit security ike gateway PARTNER_GW]
```

```
user@host# set ike-policy IKE_POL
user@host# set address 11.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=Sales
user@host# set external-interface ge-0/0/2.0
user@host# set local-address 31.1.1.2
user@host# set advpn suggester disable
user@host# set advpn partner
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals IPSEC_PROP
[edit security isec vpn PARTNER_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway PARTNER_GW
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

5. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile advpn ca-identity advpn
user@host# set ca-profile advpn enrollment url http://10.157.92.176:8080/scep/advpn/
```

6. Configure zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/4.0
user@host# set interfaces st0.1
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
```

```
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/2.0
```

7. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
    user@host# show interfaces
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 31.1.1.2/24;
            }
        }
    }
    ge-0/0/4{
        unit 0 {
            family inet {
                address 36.1.1.1/24;
            }
        }
    }
    st0 {
        unit 1 {
            multipoint;
            family inet {
                address 172.16.1.3/24;
            }
        }
    }
    [edit]
```

```
user@host# show protocols
ospf {
    graceful-restart {
        restart-duration 300;
        notify-duration 300;
        no-strict-lsa-checking;
    }
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            metric 15;
            retransmit-interval 1;
            dead-interval 40;
            demand-circuit;
            dynamic-neighbors;
        }
        interface ge-0/0/4.0;
    }
}
[edit]
user@host# show routing-options
graceful-restart;
static {
    route 11.1.1.0/24 next-hop 31.1.1.1;
    route 21.1.1.0/24 next-hop 31.1.1.1;
}
router-id 172.16.1.3;
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy IKE_POL {
    proposals IKE_PROP;
    certificate {
        local-certificate Partner2_Certificate_ID
    }
}
gateway PARTNER_GW {
    ike-policy IKE_POL;
    }
```

```
    address 11.1.1.1;
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=Sales;
    external-interface ge-0/0/2.0;
    local-address 31.1.1.2;
    advpn {
        suggester{
            disable;
        }
        partner {
        }
    }
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-256-cbc;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals IPSEC_PROP;
}
vpn PARTNER_VPN {
    bind-interface st0.1;
    ike {
        gateway PARTNER_GW;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security pki
ca-profile advpn {
    ca-identity advpn;
    enrollment {
        url http://10.157.92.176:8080/scep/advpn/;
    }
}
```

```
    [edit]
    user@host# show security zones
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/4.0;
            st0.1;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/2.0;
        }
    }
    [edit]
user@host# show security policies
    default-policy {
        permit-all;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly. First, verify that tunnels are established between the AutoVPN hub and spokes. When traffic is passed from one spoke to another through the hub, a shortcut can be established between the spokes. Verify that the shortcut partners have established a tunnel between them and that a route to the peer is installed on the partners.

Verifying Tunnels Between the Hub and Spokes

Purpose

Verify that tunnels are established between the AutoVPN hub and spokes. Initial traffic from one spoke to another must travel through the hub.

Action

From operational mode, enter the `show security ike security-associations` and `show security ipsec security-associations` commands on the hub and spokes.

The following commands are entered on the hub:

```
user@host> show security ike security-associations
node1:
--------------------------------------------------------------------------
Index    State  Initiator cookie  Responder cookie  Mode          Remote Address
10957048 UP     2d58d8fbc396762d  46145be580c68be0  IKEv2         31.1.1.2
10957049 UP     fa05ee6d0f2cfb22  16f5ca836b118c0e  IKEv2         21.1.1.2
```

```
user@host> show security ike security-associations detail
node1:
--------------------------------------------------------------------------
IKE peer 31.1.1.2, Index 10957048, Gateway Name: SUGGESTER_GW
```

```
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    0
     Suggestions accepted:    0
     Suggestions declined:    0
  Role: Responder, State: UP
  Initiator cookie: 2d58d8fbc396762d, Responder cookie: 46145be580c68be0
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 11.1.1.1:500, Remote: 31.1.1.2:500
  Lifetime: Expires in 28196 seconds
  Peer ike-id: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2030
   Output bytes  :                 2023
   Input  packets:                    4
   Output packets:                    4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 11.1.1.1:500, Remote: 31.1.1.2:500
    Local identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Remote identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created

IKE peer 21.1.1.2, Index 10957049, Gateway Name: SUGGESTER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    0
     Suggestions accepted:    0
     Suggestions declined:    0
  Role: Responder, State: UP
  Initiator cookie: fa05ee6d0f2cfb22, Responder cookie: 16f5ca836b118c0e
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 11.1.1.1:500, Remote: 21.1.1.2:500
  Lifetime: Expires in 28219 seconds
Peer ike-id: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2030
   Output bytes  :                 2023
   Input  packets:                    4
   Output packets:                    4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 11.1.1.1:500, Remote: 21.1.1.2:500
    Local identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Remote identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
node1:
--------------------------------------------------------------------------
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <201326593 ESP:aes-cbc-256/sha1 44ccf265 2999/ unlim - root 500 31.1.1.2
  >201326593 ESP:aes-cbc-256/sha1 a9d301b0 2999/ unlim - root 500 31.1.1.2
  <201326594 ESP:aes-cbc-256/sha1 98a2b155 3022/ unlim - root 500 21.1.1.2
  >201326594 ESP:aes-cbc-256/sha1 de912bcd 3022/ unlim - root 500 21.1.1.2
```

```
user@host> show security ipsec security-associations detail
node1:
--------------------------------------------------------------------------


ID: 201326593 Virtual-system: root, VPN Name: SUGGESTER_VPN
  Local Gateway: 11.1.1.1, Remote Gateway: 31.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 2, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
    Tue Jan 13 2015 12:57:48 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:57:48 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:57:48 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: 44ccf265, AUX-SPI: 0
    Hard lifetime: Expires in 2991 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2414 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: a9d301b0, AUX-SPI: 0
    Hard lifetime: Expires in 2991 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2414 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64

ID: 201326594 Virtual-system: root, VPN Name: SUGGESTER_VPN
  Local Gateway: 11.1.1.1, Remote Gateway: 21.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 3, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
    Tue Jan 13 2015 12:58:11 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:58:11 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:58:11 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: 98a2b155, AUX-SPI: 0
    Hard lifetime: Expires in 3014 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2436 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: de912bcd, AUX-SPI: 0
    Hard lifetime: Expires in 3014 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2436 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 28 destinations, 28 routes (27 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

25.1.1.0/24        *[OSPF/10] 00:00:27, metric 11
                    > to 172.16.1.2 via st0.1
36.1.1.0/24        *[OSPF/10] 00:00:27, metric 11
                    > to 172.16.1.3 via st0.1
172.16.1.2/32      *[OSPF/10] 00:00:27, metric 10
                    > to 172.16.1.2 via st0.1
172.16.1.3/32      *[OSPF/10] 00:00:27, metric 10
                    > to 172.16.1.3 via st0.1
224.0.0.5/32       *[OSPF/10] 00:00:48, metric 1
                      MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
172.16.1.3       st0.1                  Full      172.16.1.3       128     -
172.16.1.2       st0.1                  Full      172.16.1.2       128     -
```

The following commands are entered on spoke 1:

```
user@host> show security ike security-associations
node0:
--------------------------------------------------------------------------
```

```
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
578872  UP     fa05ee6d0f2cfb22  16f5ca836b118c0e  IKEv2          11.1.1.1
```

```
user@host> show security ike security-associations detail
node0:
--------------------------------------------------------------------------
IKE peer 11.1.1.1, Index 578872, Gateway Name: PARTNER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    0
     Suggestions accepted:    0
     Suggestions declined:    0
  Role: Initiator, State: UP
  Initiator cookie: fa05ee6d0f2cfb22, Responder cookie: 16f5ca836b118c0e
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 21.1.1.2:500, Remote: 11.1.1.1:500
  Lifetime: Expires in 28183 seconds
  Peer ike-id: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2023
   Output bytes  :                 2030
   Input  packets:                    4
   Output packets:                    4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 21.1.1.2:500, Remote: 11.1.1.1:500
    Local identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108866 ESP:aes-cbc-256/sha1 de912bcd 2985/ unlim - root 500 11.1.1.1
  >67108866 ESP:aes-cbc-256/sha1 98a2b155 2985/ unlim - root 500 11.1.1.1
```

```
user@host> show security ipsec security-associations detail
node0:
--------------------------------------------------------------------------


ID: 67108866 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 21.1.1.2, Remote Gateway: 11.1.1.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x8608a29
  Tunnel events:
    Tue Jan 13 2015 12:58:11 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:58:11 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:58:11 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: de912bcd, AUX-SPI: 0
    Hard lifetime: Expires in 2980 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2358 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 98a2b155, AUX-SPI: 0
    Hard lifetime: Expires in 2980 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2358 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 29 destinations, 29 routes (28 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24        *[OSPF/10] 00:11:46, metric 16
                    > to 172.16.1.1 via st0.1
36.1.1.0/24        *[OSPF/10] 00:11:46, metric 26
                    > to 172.16.1.1 via st0.1
172.16.1.1/32      *[OSPF/10] 00:11:46, metric 15
                    > to 172.16.1.1 via st0.1
172.16.1.3/32      *[OSPF/10] 00:11:46, metric 25
                    > to 172.16.1.1 via st0.1
224.0.0.5/32       *[OSPF/10] 00:16:52, metric 1
                      MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
172.16.1.1       st0.1                  Full      172.16.1.1       128     -
```

The following commands are entered on spoke 2:

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
2299162 UP     2d58d8fbc396762d  46145be580c68be0  IKEv2          11.1.1.1
```

```
user@host> show security ike security-associations detail
IKE peer 11.1.1.1, Index 2299162, Gateway Name: PARTNER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    0
     Suggestions accepted:    0
     Suggestions declined:    0
  Role: Initiator, State: UP
  Initiator cookie: 2d58d8fbc396762d, Responder cookie: 46145be580c68be0
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 31.1.1.2:500, Remote: 11.1.1.1:500
  Lifetime: Expires in 28135 seconds
  Peer ike-id: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2023
   Output bytes  :                 2030
   Input  packets:                    4
   Output packets:                    4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 31.1.1.2:500, Remote: 11.1.1.1:500
    Local identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108866 ESP:aes-cbc-256/sha1 a9d301b0 2936/ unlim - root 500 11.1.1.1
  >67108866 ESP:aes-cbc-256/sha1 44ccf265 2936/ unlim - root 500 11.1.1.1
```

```
user@host> show security ipsec security-associations detail
ID: 67108866 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 31.1.1.2, Remote Gateway: 11.1.1.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x8608a29
  Tunnel events:
    Tue Jan 13 2015 12:57:48 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:57:48 -0800: Tunnel is ready. Waiting for trigger event or peer to 
trigger negotiation (1 times)
    Tue Jan 13 2015 12:57:48 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: a9d301b0, AUX-SPI: 0
    Hard lifetime: Expires in 2933 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2311 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 44ccf265, AUX-SPI: 0
    Hard lifetime: Expires in 2933 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2311 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 36 destinations, 36 routes (35 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24        *[OSPF/10] 00:00:09, metric 16
                    > to 172.16.1.1 via st0.1
25.1.1.0/24        *[OSPF/10] 00:00:09, metric 26
                    > to 172.16.1.1 via st0.1
172.16.1.1/32      *[OSPF/10] 00:00:09, metric 15
                    > to 172.16.1.1 via st0.1
172.16.1.2/32      *[OSPF/10] 00:00:09, metric 25
                    > to 172.16.1.1 via st0.1
224.0.0.5/32       *[OSPF/10] 00:17:52, metric 1
                      MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
172.16.1.1       st0.1                  Full      172.16.1.1       128     -
```

Meaning

The show security ike security-associations command lists all active IKE Phase 1 SAs. The show security ipsec security-associations command lists all active IKE Phase 2 SAs. The hub shows two active tunnels, one to each spoke. Each spoke shows an active tunnel to the hub.

If no SAs are listed for IKE Phase 1, then there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

If no SAs are listed for IKE Phase 2, then there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

The show route protocol ospf command displays entries in the routing table that were learned from the OSPF protocol. The show ospf neighbor command displays information about OSPF neighbors.

Verifying the Shortcut Tunnel Between Partners

Purpose

The AutoVPN hub can act as a shortcut suggester when it notices that traffic is exiting a tunnel with one of its spokes and entering a tunnel with another spoke. A new IPsec SA, or shortcut, is established between the two shortcut partners. On each partner, the route to the network behind its partner now points to the shortcut tunnel instead of to the tunnel between the partner and the suggester (hub).

Action

From operational mode, enter the show security ike security-associations, show security ipsec security-associations, show route protocol ospf, and show ospf neighbor commands on the spokes.

The following commands are entered on the hub:

```
user@host> show security ike security-associations
node0:
--------------------------------------------------------------------------
Index    State  Initiator cookie  Responder cookie  Mode           Remote Address
10957048 UP     2d58d8fbc396762d  46145be580c68be0  IKEv2          31.1.1.2
10957049 UP     fa05ee6d0f2cfb22  16f5ca836b118c0e  IKEv2          21.1.1.2
```

```
user@host> show security ike security-associations detail
node0:
```

```
--------------------------------------------------------------------------
IKE peer 31.1.1.2, Index 10957048, Gateway Name: SUGGESTER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    1
     Suggestions accepted:    1
     Suggestions declined:    0
  Role: Responder, State: UP
  Initiator cookie: 2d58d8fbc396762d, Responder cookie: 46145be580c68be0
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 11.1.1.1:500, Remote: 31.1.1.2:500
  Lifetime: Expires in 27781 seconds
  Peer ike-id: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                  260
   Output bytes  :                  548
   Input  packets:                    3
   Output packets:                    3
  IPSec security associations: 0 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 11.1.1.1:500, Remote: 31.1.1.2:500
    Local identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Remote identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created

IKE peer 21.1.1.2, Index 10957049, Gateway Name: SUGGESTER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    1
     Suggestions accepted:    1
     Suggestions declined:    0
  Role: Responder, State: UP
  Initiator cookie: fa05ee6d0f2cfb22, Responder cookie: 16f5ca836b118c0e
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 11.1.1.1:500, Remote: 21.1.1.2:500
  Lifetime: Expires in 27804 seconds
  Peer ike-id: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                  244
   Output bytes  :                  548
   Input  packets:                    3
   Output packets:                    3
  IPSec security associations: 0 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 11.1.1.1:500, Remote: 21.1.1.2:500
    Local identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Remote identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
s  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <201326593 ESP:aes-cbc-256/sha1 44ccf265 2584/ unlim - root 500 31.1.1.2
  >201326593 ESP:aes-cbc-256/sha1 a9d301b0 2584/ unlim - root 500 31.1.1.2
  <201326594 ESP:aes-cbc-256/sha1 98a2b155 2607/ unlim - root 500 21.1.1.2
  >201326594 ESP:aes-cbc-256/sha1 de912bcd 2607/ unlim - root 500 21.1.1.2
```

```
user@host> show security ipsec security-associations detail
node0:
--------------------------------------------------------------------------
```

```
ID: 201326593 Virtual-system: root, VPN Name: SUGGESTER_VPN
  Local Gateway: 11.1.1.1, Remote Gateway: 31.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
    Tue Jan 13 2015 13:09:48 -0800: Bind-interface's address received. Information updated (1
times)
    Tue Jan 13 2015 13:09:48 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
  Direction: inbound, SPI: 44ccf265, AUX-SPI: 0
    Hard lifetime: Expires in 2578 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2001 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: a9d301b0, AUX-SPI: 0
    Hard lifetime: Expires in 2578 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2001 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64

ID: 201326594 Virtual-system: root, VPN Name: SUGGESTER_VPN
  Local Gateway: 11.1.1.1, Remote Gateway: 21.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
    Tue Jan 13 2015 13:09:48 -0800: Bind-interface's address received. Information updated (1
times)
    Tue Jan 13 2015 13:09:48 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
  Direction: inbound, SPI: 98a2b155, AUX-SPI: 0
    Hard lifetime: Expires in 2601 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2023 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: de912bcd, AUX-SPI: 0
    Hard lifetime: Expires in 2601 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2023 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 28 destinations, 28 routes (27 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

25.1.1.0/24        *[OSPF/10] 00:04:49, metric 11
                    > to 172.16.1.2 via st0.1
36.1.1.0/24        *[OSPF/10] 00:04:49, metric 11
                    > to 172.16.1.3 via st0.1
172.16.1.2/32      *[OSPF/10] 00:04:49, metric 10
                    > to 172.16.1.2 via st0.1
172.16.1.3/32      *[OSPF/10] 00:04:49, metric 10
                    > to 172.16.1.3 via st0.1
224.0.0.5/32       *[OSPF/10] 00:05:10, metric 1
                      MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
172.16.1.3       st0.1                  Full      172.16.1.3       128     -
172.16.1.2       st0.1                  Full      172.16.1.2       128     -
```

The following commands are entered on spoke 1:

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
```

```
578872  UP     fa05ee6d0f2cfb22  16f5ca836b118c0e  IKEv2          11.1.1.1
578873  UP     895e4d9c7c5da7a4  17de7f18b45139b4  IKEv2          31.1.1.2
```

```
user@host> show security ike security-associations detail
node0:
--------------------------------------------------------------------------
IKE peer 11.1.1.1, Index 578872, Gateway Name: PARTNER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    1
     Suggestions accepted:    1
     Suggestions declined:    0
  Role: Initiator, State: UP
  Initiator cookie: fa05ee6d0f2cfb22, Responder cookie: 16f5ca836b118c0e
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 21.1.1.2:500, Remote: 11.1.1.1:500
  Lifetime: Expires in 27906 seconds
  Peer ike-id: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2495
   Output bytes  :                 2274
   Input  packets:                    6
   Output packets:                    7
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 21.1.1.2:500, Remote: 11.1.1.1:500
    Local identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Flags: IKE SA is created

IKE peer 31.1.1.2, Index 578873, Gateway Name: PARTNER_GW
```

```
  Auto Discovery VPN:
   Type: Shortcut, Local Capability: Partner, Peer Capability: Partner
  Role: Initiator, State: UP
  Initiator cookie: 895e4d9c7c5da7a4, Responder cookie: 17de7f18b45139b4
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 21.1.1.2:500, Remote: 31.1.1.2:500
  Lifetime: Expires in 28787 seconds
  Peer ike-id: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 1855
   Output bytes  :                 1990
   Input  packets:                    2
   Output packets:                    2
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 21.1.1.2:500, Remote: 31.1.1.2:500
    Local identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108866 ESP:aes-cbc-256/sha1 de912bcd 2709/ unlim - root 500 11.1.1.1
  >67108866 ESP:aes-cbc-256/sha1 98a2b155 2709/ unlim - root 500 11.1.1.1
  <67108868 ESP:aes-cbc-256/sha1 75d0177b 3590/ unlim - root 500 31.1.1.2
  >67108868 ESP:aes-cbc-256/sha1 e4919d73 3590/ unlim - root 500 31.1.1.2
```

```
user@host> show security ipsec security-associations detail
node0:
--------------------------------------------------------------------------

ID: 67108866 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 21.1.1.2, Remote Gateway: 11.1.1.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x8608a29
  Tunnel events:
    Tue Jan 13 2015 12:58:11 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:58:11 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:58:11 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: de912bcd, AUX-SPI: 0
    Hard lifetime: Expires in 2701 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2079 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 98a2b155, AUX-SPI: 0
    Hard lifetime: Expires in 2701 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2079 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64

ID: 67108868 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 21.1.1.2, Remote Gateway: 31.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Auto Discovery VPN:
    Type: Shortcut, Shortcut Role: Initiator
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x40608a29
  Tunnel events:
    Tue Jan 13 2015 13:12:52 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 13:12:52 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 13:12:52 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: 75d0177b, AUX-SPI: 0
    Hard lifetime: Expires in 3582 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2959 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: e4919d73, AUX-SPI: 0
    Hard lifetime: Expires in 3582 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2959 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 29 destinations, 29 routes (28 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24        *[OSPF/10] 00:03:29, metric 16
                    > to 172.16.1.1 via st0.1
36.1.1.0/24        *[OSPF/10] 00:00:35, metric 16
                    > to 172.16.1.3 via st0.1
172.16.1.1/32      *[OSPF/10] 00:03:29, metric 15
                    > to 172.16.1.1 via st0.1
172.16.1.3/32      *[OSPF/10] 00:00:35, metric 15
                    > to 172.16.1.3 via st0.1
224.0.0.5/32       *[OSPF/10] 00:20:22, metric 1
                      MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
```

```
172.16.1.3       st0.1                 Full      172.16.1.3       128     -
172.16.1.1       st0.1                 Full      172.16.1.1       128
```

The following commands are entered on spoke 2:

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
2299162 UP     2d58d8fbc396762d  46145be580c68be0  IKEv2          11.1.1.1
2299163 UP     895e4d9c7c5da7a4  17de7f18b45139b4  IKEv2          21.1.1.2
```

```
user@host> show security ike security-associations detail
IKE peer 11.1.1.1, Index 2299162, Gateway Name: PARTNER_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    1
     Suggestions accepted:    1
     Suggestions declined:    0
  Role: Initiator, State: UP
  Initiator cookie: 2d58d8fbc396762d, Responder cookie: 46145be580c68be0
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 31.1.1.2:500, Remote: 11.1.1.1:500
  Lifetime: Expires in 27835 seconds
  Peer ike-id: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2571
   Output bytes  :                 2290
   Input  packets:                    7
   Output packets:                    7
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 31.1.1.2:500, Remote: 11.1.1.1:500
    Local identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=suggester, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
    Flags: IKE SA is created

IKE peer 21.1.1.2, Index 2299163, Gateway Name: PARTNER_GW
  Auto Discovery VPN:
   Type: Shortcut, Local Capability: Partner, Peer Capability: Partner
  Role: Responder, State: UP
  Initiator cookie: 895e4d9c7c5da7a4, Responder cookie: 17de7f18b45139b4
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 31.1.1.2:500, Remote: 21.1.1.2:500
  Lifetime: Expires in 28739 seconds
  Peer ike-id: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 2066
   Output bytes  :                 1931
   Input  packets:                    3
   Output packets:                    3
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 31.1.1.2:500, Remote: 21.1.1.2:500
    Local identity: DC=XYZ, CN=partner2, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Remote identity: DC=XYZ, CN=partner1, OU=Sales, O=XYZ, L=NewYork, ST=NY, C=US
    Flags: IKE SA is created
```

```
user@host> show security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108866 ESP:aes-cbc-256/sha1 a9d301b0 2638/ unlim - root 500 11.1.1.1
  >67108866 ESP:aes-cbc-256/sha1 44ccf265 2638/ unlim - root 500 11.1.1.1
  <67108868 ESP:aes-cbc-256/sha1 e4919d73 3542/ unlim - root 500 21.1.1.2
  >67108868 ESP:aes-cbc-256/sha1 75d0177b 3542/ unlim - root 500 21.1.1.2
```

```
user@host> show security ipsec security-associations detail
ID: 67108866 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 31.1.1.2, Remote Gateway: 11.1.1.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x8608a29
  Tunnel events:
    Tue Jan 13 2015 12:57:48 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 12:57:48 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 12:57:48 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: a9d301b0, AUX-SPI: 0
    Hard lifetime: Expires in 2632 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2010 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 44ccf265, AUX-SPI: 0
    Hard lifetime: Expires in 2632 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2010 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64

ID: 67108868 Virtual-system: root, VPN Name: PARTNER_VPN
  Local Gateway: 31.1.1.2, Remote Gateway: 21.1.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Auto Discovery VPN:
    Type: Shortcut, Shortcut Role: Responder
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x40608aa9
  Tunnel events:
    Tue Jan 13 2015 13:12:52 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Jan 13 2015 13:12:52 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Jan 13 2015 13:12:52 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: e4919d73, AUX-SPI: 0
    Hard lifetime: Expires in 3536 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2958 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 75d0177b, AUX-SPI: 0
    Hard lifetime: Expires in 3536 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2958 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

```
user@host> show route protocol ospf
inet.0: 36 destinations, 36 routes (35 active, 0 holddown, 1 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24        *[OSPF/10] 00:03:55, metric 16
                    > to 172.16.1.1 via st0.1
25.1.1.0/24        *[OSPF/10] 00:01:02, metric 16
                    > to 172.16.1.2 via st0.1
172.16.1.1/32      *[OSPF/10] 00:03:55, metric 15
                    > to 172.16.1.1 via st0.1
172.16.1.2/32      *[OSPF/10] 00:01:02, metric 15
                    > to 172.16.1.2 via st0.1
224.0.0.5/32       *[OSPF/10] 00:21:38, metric 1
                      MultiRecv
```

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
172.16.1.2       st0.1                  Full      172.16.1.2       128     -
172.16.1.1       st0.1                  Full      172.16.1.1       128     -
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. The hub still shows two active tunnels, one to each spoke. Each spoke shows two active tunnels, one to the hub and one to its shortcut partner.

The `show route protocol ospf` command shows the addition of routes to the partner and to the hub.

SEE ALSO

Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls | 199

Example: Configuring ADVPN with OSPFv3 for IPv6 Traffic

IN THIS SECTION

This example shows how to configure an ADVPN hub and two spokes to create a shortcut tunnel and change the routing topology for the host to reach the other side without sending traffic through the hub. This example configures ADVPN for IPv6 environment using OSPFv3 to forward packets through the VPN tunnels.

Requirements

This example uses the following hardware and software components:

- Three supported SRX Series Firewalls as ADVPN hub and spokes
- Junos OS Release 18.1R1, and later releases.

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

This example shows the configuration of an ADVPN hub and the subsequent configurations of two spokes.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). The certificates for the spokes contain the organizational unit (OU) value “SLT” in the subject field; the hub is configured with a group IKE ID to match the value “SLT” in the OU field.

The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the ADVPN hub and all spokes must have the same values. Table 86 on page 982 shows the options used in this example.

Table 86: Phase 1 and Phase 2 Options for ADPN Hub and Spoke Basic OSPFv3 Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	19
Authentication algorithm	SHA-384
Encryption algorithm	AES 256 CBC
IKE policy:	

Table 86: Phase 1 and Phase 2 Options for ADPN Hub and Spoke Basic OSPFv3 Configurations *(Continued)*

Option	Value
Mode	Main
IPsec proposal:	
Protocol	ESP
Lifetime seconds	3000
Encryption algorithm	AES 256 GCM
IPsec policy:	
Perfect Forward Secrecy (PFS) group	19

The same certificate authority (CA) is configured on all devices.

Table 87 on page 983 shows the options configured on the hub and on all spokes.

Table 87: ADVPN OSPFv3 Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	2001:db8:2000::1
Remote IKE ID	Distinguished name (DN) on the spoke’s certificate with the string `SLT` in the organizational unit (OU) field	DN on the hub’s certificate
Local IKE ID	DN on the hub’s certificate	DN on the spoke’s certificate

Table 87: ADVPN OSPFv3 Configuration for Hub and All Spokes *(Continued)*

Option	Hub	All Spokes
External interface	reth1	Spoke 1: ge-0/0/0.0 Spoke 2: ge-0/0/0.0
VPN:		
Bind interface	st0.1	st0.1
Establish tunnels	(not configured)	establish-tunnels immediately

Table 88 on page 984 shows the configuration options that are different on each spoke.

Table 88: Comparison Between the OSPFv3 Spoke Configurations

Option	Spoke 1	Spoke 2
st0.1 interface	2001:db8:9000::2/64	2001:db8:9000::3/64
Interface to internal network	(ge-0/0/1.0) 2001:db8:4000::1/64	(ge-0/0/1.0) 2001:db8:6000::1/64
Interface to Internet	(ge-0/0/0.0) 2001:db8:3000::2/64	(ge-0/0/0.0) 2001:db8:5000::2/64

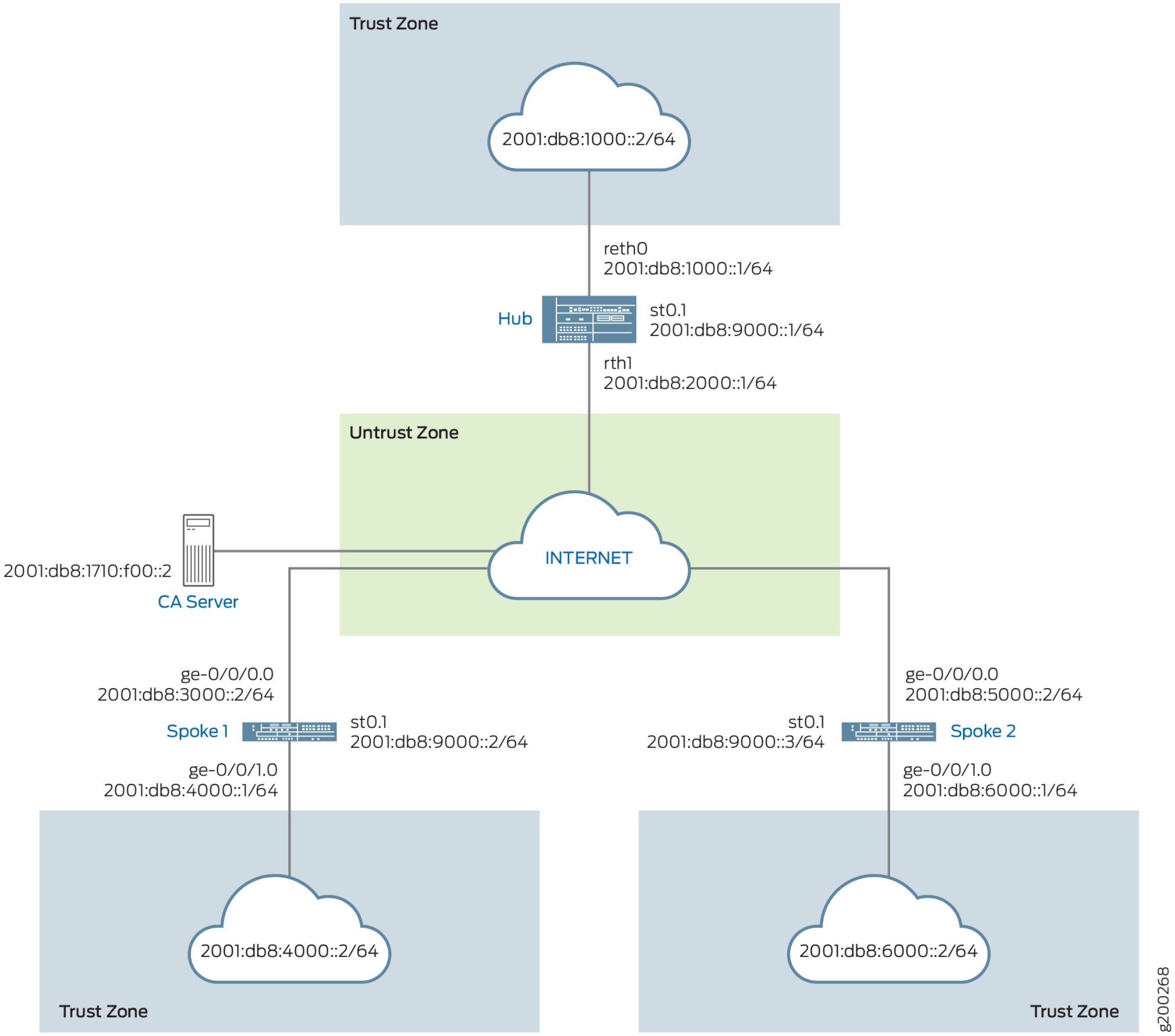
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Topology

Figure 65 on page 985 shows the SRX Series Firewalls to be configured for ADVPN in this example.

Figure 65: ADVPN Deployment with OSPFv3



Configuration

IN THIS SECTION

To configure ADVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 2:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke2@example.net ip-address 10.3.3.1 subject
DC=example.net,CN=spoke2,OU=SLT,O=example,L=Tumkur,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40bb71d400000000258f
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Tumkur, Common name: spoke2, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
  Alternate subject: "spoke2@example.net", example.net, 10.3.3.1
  Validity:
    Not before: 11- 6-2012 10:02
    Not after: 11- 6-2013 10:12
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:b6:2e:e2:da:e6:ac:57:e4:5d:ff:de:f6:89
    27:d6:3e:1b:4a:3f:b2:2d:b3:d3:61:ed:ed:6a:07:d9:8a:d2:24:03
    77:1a:fe:84:e1:12:8a:2d:63:6e:bf:02:6b:15:96:5a:4f:37:a0:46
    44:09:96:c0:fd:bb:ab:79:2c:5d:92:bd:31:f0:3b:29:51:ce:89:8e
    7c:2b:02:d0:14:5b:0a:a9:02:93:21:ea:f9:fc:4a:e7:08:bc:b1:6d
    7c:f8:3e:53:58:8e:f1:86:13:fe:78:b5:df:0b:8e:53:00:4a:46:11
    58:4a:38:e9:82:43:d8:25:47:7d:ef:18:f0:ef:a7:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    1a:6d:77:ac:fd:94:68:ce:cf:8a:85:f0:39:fc:e0:6b:fd:fe:b8:66 (sha1)
    00:b1:32:5f:7b:24:9c:e5:02:e6:72:75:9e:a5:f4:77 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set chassis cluster reth-count 2
set chassis cluster node 0
set chassis cluster node 1
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 254
set chassis cluster redundancy-group 1 node 1 priority 1
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate HUB
set security ike gateway IKE_GWA_1 ike-policy IKE_POL
set security ike gateway IKE_GWA_1 dynamic distinguished-name wildcard OU=SLT
set security ike gateway IKE_GWA_1 dynamic ike-user-type group-ike-id
set security ike gateway IKE_GWA_1 dead-peer-detection always-send
set security ike gateway IKE_GWA_1 dead-peer-detection interval 10
set security ike gateway IKE_GWA_1 dead-peer-detection threshold 3
set security ike gateway IKE_GWA_1 local-identity distinguished-name
set security ike gateway IKE_GWA_1 external-interface reth1
set security ike gateway IKE_GWA_1 advpn partner disable
set security ike gateway IKE_GWA_1 version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPNA_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPNA_1 ike gateway IKE_GWA_1
set security ipsec vpn IPSEC_VPNA_1 ike ipsec-policy IPSEC_POL
set security policies default-policy permit-all
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces reth0.0
set interfaces ge-0/0/0 gigether-options redundant-parent reth1
set interfaces ge-0/0/1 gigether-options redundant-parent reth0
set interfaces ge-7/0/0 gigether-options redundant-parent reth1
set interfaces ge-7/0/1 gigether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet
set interfaces reth0 unit 0 family inet6 address 2001:db8:1000::1/64
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet
set interfaces reth1 unit 0 family inet6 address 2001:db8:2000::1/64
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet6 address 2001:db8:9000::1/64
set routing-options rib inet6.0 static route 2001:db8:3000::0/64 next-hop 2001:db8:2000::2
set routing-options rib inet6.0 static route 2001:db8:5000::0/64 next-hop 2001:db8:2000::2
set protocols ospf3 area 0.0.0.0 interface reth0.0
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 gigether-options redundant-parent reth1
user@host# set ge-0/0/1 gigether-options redundant-parent reth0
user@host# set ge-7/0/0 gigether-options redundant-parent reth1
user@host# set ge-7/0/1 gigether-options redundant-parent reth0
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet
user@host# set reth0 unit 0 family inet6 address 2001:db8:1000::1/64
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet
user@host# set reth1 unit 0 family inet6 address 2001:db8:2000::1/64
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet6 address 2001:db8:9000::1/64
```

2. Configure the routing protocol.

```
[edit protocols ospf3]
user@host# set ospf3 area 0.0.0.0 interface reth0.0
user@host# set ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:3000::0/64 next-hop 2001:db8:2000::2
user@host# set rib inet6.0 static route 2001:db8:5000::0/64 next-hop 2001:db8:2000::2
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate HUB
[edit security ike gateway IKE_GWA_1]
user@host# set ike-policy IKE_POL
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set ike-user-type group-ike-id
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set external-interface reth1
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPNA_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GWA_1
user@host# set ike ipsec-policy IPSEC_POL
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces reth1.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces reth0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set pki ca-profile ROOT-CA revocation-check disable
```

8. Configure chassis cluster

```
[edit chassis cluster]
set reth-count 2
set node 0
set node 1
set redundancy-group 0 node 0 priority 254
set redundancy-group 0 node 1 priority 1
set redundancy-group 1 node 0 priority 254
set redundancy-group 1 node 1 priority 1
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` `show chassis cluster` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-0/0/1 {
    gigether-options {
        redundant-parent reth0;
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet;
            family inet6 {
                address 2001:db8:1000::1/64;
            }
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet;
            family inet6 {
                address 2001:db8:2000::1/64;
            }
        }
    }
    st0 {
        unit 1 {
            multipoint;
            family inet6 {
                address 2001:db8:9000::1/64 {
                    primary;
                }
            }
        }
    }
[edit]
user@host# show protocols
ospf3 {
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            demand-circuit;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
        interface reth0.0;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
        route 2001:db8:3000::/64 next-hop 2001:db8:2000::2;
        route 2001:db8:5000::/64 next-hop 2001:db8:2000::2;
    }
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate HUB;
    }
}
gateway IKE_GWA_1 {
    ike-policy IKE_POL;
    dynamic {
        distinguished-name {
            wildcard OU=SLT;
        }
        ike-user-type group-ike-id;
    }
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    external-interface reth1;
    advpn {
        partner {
            disable;
        }
    }
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPNA_1 {
    bind-interface st0.1;
    ike {
        gateway IKE_GWA_1;
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        st0.1;
        reth1.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        reth0.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
```

```
            retry-interval 0;
        }
        revocation-check {
            disable;
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE1
set security ike gateway IKE_GW_SPOKE_1 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_1 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_1 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_1 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_1 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_1 advpn suggester disable
set security ike gateway IKE_GW_SPOKE_1 version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike gateway IKE_GW_SPOKE_1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_1 establish-tunnels immediately
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/1.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet6 address 2001:db8:9000::2/64
set routing-options rib inet6.0 static route 2001:db8:2000::0/64 next-hop 2001:db8:3000::1
set protocols ospf3 area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 1:

1. Configure interfaces.

   ```
   [edit interfaces]
   user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
   user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
   user@host# set st0 unit 1 multipoint
   user@host# set st0 unit 1 family inet6 address 2001:db8:9000::2/64
   ```

2. Configure the routing protocol.

```
[edit protocols ospf3]
set area 0.0.0.0 interface ge-0/0/1.0
set area 0.0.0.0 interface st0.1 interface-type p2mp
set area 0.0.0.0 interface st0.1 dynamic-neighbors
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:3000::1
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE1
[edit security ike gateway IKE_GW_SPOKE_1]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0.0
user@host# set advpn suggester disable
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROPl]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
```

```
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPN_SPOKE_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GW_SPOKE_1
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show` security pki commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:3000::2/64;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:db8:4000::1/64;
        }
    }
}
st0 {
    unit 1 {
        multipoint;
        family inet6 {
            address 2001:db8:9000::2/64;
        }
    }
}
[edit]
user@host# show protocols
ospf3 {
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
        route 2001:db8:2000::/64 next-hop [ 2001:db8:3000::1 2001:db8:5000::1 ];
    }
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE1;
    }
}
gateway IKE_GW_SPOKE_1 {
    ike-policy IKE_POL;
    address 2001:db8:2000::1;
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=SLT;
    external-interface ge-0/0/0.0;
    advpn {
        suggester {
            disable;
        }
    }
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
```

```
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPN_SPOKE_1 {
    bind-interface st0.1;
    ike {
        gateway IKE_GW_SPOKE_1;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        st0.1;
        ge-0/0/0.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
         ge-0/0/1.0;
    }
```

```
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE2
set security ike gateway IKE_GW_SPOKE_2 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_2 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_2 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_2 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_2 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_2 advpn suggester disable
set security ike gateway IKE_GW_SPOKE_2 version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_2 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike gateway IKE_GW_SPOKE_2
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_2 establish-tunnels immediately
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/1.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
set interfaces st0 unit 1 family inet6 address 2001:db8:9000::3/64
set routing-options rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
set protocols ospf3 area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
user@host# set st0 unit 1 family inet6 address 2001:db8:9000::3/64
```

2. Configure the routing protocol.

```
[edit protocols ospf3]
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/1.0
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE2
[edit security ike gateway IKE_GW_SPOKE_2]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0.0
```

```
user@host# set advpn suggester disable
user@host# set version v2-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP1]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPN_SPOKE_2]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GW_SPOKE_2
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
```

```
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:5000::2/64;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:db8:6000::1/64;
        }
    }
}
    st0 {
        unit 1 {
            family inet6 {
                address 2001:db8:9000::3/64;
            }
        }
    }
[edit]
user@host# show protocols
ospf3 {
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
    route 2001:db8:2000::/64 next-hop [ 2001:db8:3000::1 2001:db8:5000::1 ];
    }
}
[edit]
user@host# show security ike
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE2;
    }
}
gateway IKE_GW_SPOKE_2 {
    ike-policy IKE_POL;
    address 2001:db8:2000::1;
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=SLT;
    external-interface ge-0/0/0.0;
    advpn {
        suggester {
        disable
        }
    }
    version v2-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPN_SPOKE_2 {
    bind-interface st0.1;
    ike {
        gateway IKE_GW_SPOKE_2;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/0.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                ospf3;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying IKE Status

Purpose

Verify the IKE status.

Action

From operational mode, enter the **show security ike sa** command.

```
user@host> show security ike sa
Index   State Initiator cookie          Responder cookie          Mode Remote Address


4295070 UP    2001:db8:1ad4ba7a115fa229 2001:db8:32e6382a058bb296 Main 2001:db8:3000::2


295069  UP    2001:db8:88a1520c20cbbe04 2001:db8:7fa4c8e365393c48 Main 2001:db8:5000::2
```

Meaning

The `show security ike sa` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

Verifying IPsec Status

Purpose

Verify the IPsec status.

Action

From operational mode, enter the **show security ipsec sa** command.

```
user@host> show security ipsec sa
Total active tunnels: 2     Total Ipsec sas: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <67108881 ESP:aes-gcm-256/None 3dba3f80 2979/ unlim - root 500 2001:db8:5000::2
  >67108881 ESP:aes-gcm-256/None 46746d5d 2979/ unlim - root 500 2001:db8:5000::2
  <67108882 ESP:aes-gcm-256/None 16dceb60 2992/ unlim - root 500 2001:db8:3000::2
  >67108882 ESP:aes-gcm-256/None 681209c2 2992/ unlim - root 500 2001:db8:3000::2
```

Meaning

The `show security ipsec sa` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway              interface  IPSec VPN name Flag IKE-ID
XAUTH username
2001:db8:9000::2              st0.1      IPSEC_VPNA_1   Auto C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available
2001:db8:9000::3              st0.1      IPSEC_VPNA_1   Auto C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available
2001:db8::5668:ad10:fcd8:10c8 st0.1      IPSEC_VPNA_1   Auto C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available
2001:db8::5668:ad10:fcd8:112f st0.1      IPSEC_VPNA_1   Auto C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying OSPFv3

Purpose

Verify that OSPFv3 references the IP addresses for the `st0` interfaces of the spokes.

Action

From operational mode, enter the **show ospf3 neighbor interface** command.

```
user@host> show ospf3 neighbor interface
ID                        Interface              State    Pri   Dead
2001:db8:9000:2   st0.1                  Full     128      -
  Neighbor-address 2001:db8::5668:ad10:fcd8:110e


2001:db8:20:54:49.693           INFO     ${ret} = ID      Interface  State Pri Dead
2001:db8:9000:3   st0.1                  Full     128      -
  Neighbor-address 2001:db8::5668:ad10:fcd8:110e
```

SEE ALSO

Example: Configuring a Route-Based VPN | 395

Enabling OSPF to Update Routes Quickly After ADVPN Shortcut Tunnels Are Established

IN THIS SECTION

Problem

Description

OSPF can take up to 9 seconds to update a shortcut route in the routing table. It can take up to 10 seconds before traffic is forwarded to the shortcut tunnel.

Symptoms

When a shortcut tunnel is established between two shortcut partners, OSPF initiates an OSPF hello packet. Because of the timing of the shortcut tunnel establishment and the OSPF neighbor installation, the first packet in the tunnel might be dropped. This can cause OSPF to try again to establish an OSPF adjacency.

By default, the interval at which the OSPF retries to establish an adjacency is 10 seconds. After a shortcut tunnel is established, it can take more than 10 seconds for OSPF to establish an adjacency between the partners.

Solution

Configuring a smaller retry interval, such as 1 or 2 seconds, can enable OSPF to establish adjacencies faster over the shortcut tunnel. For example, use the following configurations:

```
[edit]
set protocols ospf area 0.0.0.0 interface st0.1 retransmit-interval 1
set protocols ospf area 0.0.0.0 interface st0.1 dead-interval 40
```

SEE ALSO

Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls | 199

Release History Table

Release	Description
19.2R1	Starting in Junos OS Release 19.2R1, on SRX300, SRX320, SRX340, SRX345, SRX550, SRX1500, vSRX Virtual Firewall 2.0 (with 2 vCPUs), and vSRX Virtual Firewall 3.0 (with 2 vCPUs) Series devices, Protocol Independent Multicast (PIM) using point-to-multipoint (P2MP) mode supports Auto Discovery VPN in which a new `p2mp` interface type is introduced for PIM.
18.1R1	Starting with Junos OS Release 18.1R1, ADVPN supports IPv6.

RELATED DOCUMENTATION

IPv6 IPsec VPNs | **349**

12

CHAPTER

AutoVPN

AutoVPN on Hub-and-Spoke Devices

IN THIS SECTION

AutoVPN supports an IPsec VPN aggregator (known as a hub) that serves as a single termination point for multiple tunnels to remote sites (known as spokes). AutoVPN allows network administrators to configure a hub for current and future spokes.

Understanding AutoVPN

IN THIS SECTION

- Understanding AutoVPN with Traffic Selectors | 1025

AutoVPN supports an IPsec VPN aggregator (known as a *hub*) that serves as a single termination point for multiple tunnels to remote sites (known as *spokes*). AutoVPN allows network administrators to configure a hub for current and future spokes. No configuration changes are required on the hub when spoke devices are added or deleted, thus allowing administrators flexibility in managing large-scale network deployments.

Secure Tunnel Modes

AutoVPN is supported on route-based IPsec VPNs. For route-based VPNs, you configure a secure tunnel (st0) interface and bind it to an IPsec VPN tunnel. st0 interfaces in AutoVPN networks can be configured in one of two modes:

- Point-to-point mode—By default, a st0 interface configured at the [`edit interfaces st0 unit` *x*] hierarchy level is in point-to-point mode. Starting with Junos OS Release 17.4R1, IPv6 address is supported on AutoVPN.

- Point-to-multipoint mode—In this mode, the `multipoint` option is configured at the [`edit interfaces st0 unit` *x*] hierarchy level on both AutoVPN hub and spokes. st0 interfaces on the hub and spokes must be numbered and the IP address configured on a spoke must exist in the hub's st0 interface subnetwork.

Table 89 on page 1022 compares AutoVPN point-to-point and point-to-multipoint secure tunnel interface modes.

Table 89: Comparison Between AutoVPN Point-to-Point and Point-to-Multipoint Secure Tunnel Modes

Point-to-Point Mode	Point-to-Multipoint Mode
Supports IKEv1 or IKEv2.	Supports IKEv1 or IKEv2.
Supports IPv4 and IPv6 traffic.	Supports IPv4 or IPv6.
Traffic selectors	Dynamic routing protocols (OSPF, OSPFv3 and iBGP)
Dead peer detection	Dead peer detection

Table 89: Comparison Between AutoVPN Point-to-Point and Point-to-Multipoint Secure Tunnel Modes *(Continued)*

Point-to-Point Mode	Point-to-Multipoint Mode
Allows spoke devices to be SRX Series or third-party devices.	This mode is only supported with SRX Series Firewalls.

Authentication

The supported authentication for AutoVPN hubs and spokes is X.509 public key infrastructure (PKI) certificates. The group IKE user type configured on the hub allows strings to be specified to match the alternate subject field in spoke certificates. Partial matches for the subject fields in spoke certificates can also be specified. See "Understanding Spoke Authentication in AutoVPN Deployments" on page 1027.

Starting in Junos OS Release 21.2R1, SRX5000 line with SPC3 card and vSRX Virtual Firewall running iked process supports AutoVPN with seeded preshared key. The SRX5000 line with a SPC3 card and vSRX Virtual Firewall supports AutoVPN PSK only if the junos-ike-package is installed.

We support AutoVPN with the following two options:

- **Auto-VPN seeded PSK**: Multiple peers connecting to same gateway having different pre-shared key.
- **Auto-VPN shared PSK**: Multiple peers connecting to same gateway having same pre-shared key.

Seeded PSK is different from non-seeded PSK (that is, same shared PSK). Seeded PSK uses master key to generate the shared PSK for the peer. So each peer will have different PSK connecting to the same gateway. For example: Consider a scenario where peer 1 with the IKE ID *user1@juniper.net* and peer 2 with IKE ID *user2@juniper.net* attempts to connect to gateway. In this scenario the gateway that is configured as `HUB_GW` containing the master key configured as `ThisIsMySecretPreSharedkey` will have the different PSK as follows:

Peer 1 : `79e4ea39f5c06834a3c4c031e37c6de24d46798a`

Peer 2: `3db8385746f3d1e639435a882579a9f28464e5c7`

This means, for different users with different user id and same master key will generate a different or unique preshared key.

You can use either `seeded-pre-shared-key` or `pre-shared-key` for Auto-VPN PSK:

- **Different preshared key**: If the `seeded-pre-shared-key` is set, different IKE preshared key is used by the VPN gateway to authenticate each remote peer. The peer preshared keys are generated using the `master-key` set in the IKE gateway and shared across the peers.

To enable the VPN gateway to use a different IKE preshared key (PSK) for authenticating each remote peer, use the new CLI commands `seeded-pre-shared-key` *`ascii-text`* or `seeded-pre-shared-key` *`hexadecimal`* under the `[edit security ike policy` *`policy_name`*`]` hierarchy level.

This command is mutually exclusive with `pre-shared-key` command under the same hierarchy.

See "policy" on page 1579.

- **Shared/Same preshared key**: If `pre-shared-key-type` is not configured, then the PSK is considered to be shared. Same IKE preshared key is used by the VPN gateway to authenticate all remote peers.

  To enable the VPN gateway to use the same IKE PSK for authenticating all remote peers, use the existing CLI commands `pre-sharedkey` *`ascii-text`* or `pre-shared-key` *`hexadecimal`*.

At the VPN gateway, you can bypass the IKE ID validation using the `general-ikeid` configuration statement under the `[edit security ike gateway` *`gateway_name`* `dynamic]` hierarchy level. If this option is configured, then during authentication of remote peer, the VPN gateway allows any remote IKE ID connection. See "general-ikeid" on page 1511.

The SRX5000 line with SPC3 card and vSRX Virtual Firewall running iked supports the following IKE modes:

Table 90: AutoVPN PSK Support

IKE Mode	SRX5000 Line with SPC3 Card and vSRX Virtual Firewall running iked process	
	Shared PSK	Seeded-PSK
IKEv2	Yes	Yes
IKEv2 with any-`remote-id`	Yes	Yes
IKEv1 Aggressive Mode	Yes	Yes
IKEv1 Aggressive Mode with `any-remote-id/general-ikeid`	Yes	Yes
IKEv1 main mode	Yes	No

Table 90: AutoVPN PSK Support *(Continued)*

IKE Mode	SRX5000 Line with SPC3 Card and vSRX Virtual Firewall running iked process	
	Shared PSK	Seeded-PSK
IKEv1 main mode with any-remote-id/`general-ikeid`	Yes	No

See "Example: Configuring AutoVPN with Pre-Shared Key" on page 1294.

Configuration and Management

AutoVPN is configured and managed on SRX Series Firewalls using the CLI. Multiple AutoVPN hubs can be configured on a single SRX Series Firewall. The maximum number of spokes supported by a configured hub is specific to the model of the SRX Series Firewall.

Understanding AutoVPN Limitations

The following features are not supported for AutoVPN:

- Policy-based VPNs are not supported.
- The RIP dynamic routing protocol is not supported with AutoVPN tunnels.
- Manual keys and Autokey IKE with preshared keys are not supported.
- Configuring static next-hop tunnel binding (NHTB) on the hub for spokes is not supported.
- Multicast is not supported.
- The group IKE ID user type is not supported with an IP address as the IKE ID.
- When the group IKE ID user type is used, the IKE ID should not overlap with other IKE gateways configured on the same external interface.

Understanding AutoVPN with Traffic Selectors

AutoVPN hubs can be configured with multiple traffic selectors to protect traffic to spokes. This feature provides the following benefits:

- A single VPN configuration can support many different peers.

- VPN peers can be non-SRX Series Firewalls.
- A single peer can establish multiple tunnels with the same VPN.
- A larger number of tunnels can be supported than with AutoVPN with dynamic routing protocols.

Starting with Junos OS Release 17.4R1, AutoVPN networks that use secure tunnel interfaces in point-to-point mode support IPv6 addresses for traffic selectors and for IKE peers.

When the hub-to-spoke tunnel is established, the hub uses *auto route insertion (ARI)*, known in previous releases as *reverse route insertion (RRI)*, to insert the route to the spoke prefix in its routing table. The ARI route can then be imported to routing protocols and distributed to the core network.

AutoVPN with traffic selectors can be configured with the secure tunnel (st0) interface in point-to-point mode for both IKEv1 and IKEv2.

Dynamic routing protocols are not supported on st0 interfaces when traffic selectors are configured.

Note the following caveats when configuring AutoVPN with traffic selectors:

- Dynamic routing protocols are not supported with traffic selectors with st0 interfaces in point-to-point mode.
- Auto Discovery VPN and IKEv2 configuration payload cannot be configured with AutoVPN with traffic selectors.
- Spokes can be non-SRX Series Firewalls; however, note the following differences:
    - In IKEv2, a non-SRX Series spoke can propose multiple traffic selectors in a single SA negotiation. This is not supported on SRX Series Firewalls and the negotiation is rejected.
    - A non-SRX Series spoke can identify specific ports or protocols for traffic selector use. Ports and protocols are not supported with traffic selectors on SRX Series Firewalls and the negotiation is rejected.

SEE ALSO

Understanding Spoke Authentication in AutoVPN Deployments | 1027

Understanding Traffic Selectors in Route-Based VPNs | 524

Example: Configuring Traffic Selectors in a Route-Based VPN | 531

Understanding Spoke Authentication in AutoVPN Deployments

IN THIS SECTION

In AutoVPN deployments, the hub and spoke devices must have valid X.509 PKI certificates loaded. You can use the `show security pki local-certificate detail` command to display information about the certificates loaded in a device.

This topic covers the configuration on the hub that allows spokes to authenticate and connect to the hub:

Group IKE ID Configuration on the Hub

The group IKE ID feature allows a number of spoke devices to share an IKE configuration on the hub. The certificate holder's identification, in the subject or alternate subject fields in each spoke's X.509 certificate, must contain a part that is common to all spokes; the common part of the certificate identification is specified for the IKE configuration on the hub.

For example, the IKE ID `example.net` can be configured on the hub to identify spokes with the hostnames `device1.example.net`, `device2.example.net`, and `device3.example.net`. The certificate on each spoke must contain a hostname identity in the alternate subject field with `example.net` in the right-most part of the field; for example, `device1.example.net`. In this example, all spokes use this hostname identity in their IKE ID payload. During IKE negotiation, the IKE ID from a spoke is used to match the common part of the peer IKE identity configured on the hub. A valid certificate authenticates the spoke.

The common part of the certificate identification can be one of the following:

- A partial hostname in the right-most part of the alternate subject field of the certificate, for example `example.net`.
- A partial e-mail address in the right-most part of the alternate subject field of the certificate, for example `@example.net`.
- A container string, a set of wildcards, or both to match the subject fields of the certificate. The subject fields contain details of the digital certificate holder in Abstract Syntax Notation One (ASN.1) distinguished name (DN) format. Fields can include organization, organizational unit, country, locality, or common name.

To configure a group IKE ID to match subject fields in certificates, you can specify the following types of identity matches:

- Container—The hub authenticates the spoke’s IKE ID if the subject fields of the spoke’s certificate exactly match the values configured on the hub. Multiple entries can be specified for each subject field (for example, `ou=eng,ou=sw`). The order of values in the fields must match.

- Wildcard—The hub authenticates the spoke’s IKE ID if the subject fields of the spoke’s certificate match the values configured on the hub. The wildcard match supports only one value per field (for example, `ou=eng` or `ou=sw` but not `ou=eng,ou=sw`). The order of the fields is inconsequential.

The following example configures a group IKE ID with the partial hostname `example.net` in the alternate subject field of the certificate.

```
[edit]
security {
    ike {
        policy common-cert-policy {
            proposals common-ike-proposal;
            certificate {
                local-certificate hub-local-certificate;
            }
        }
        gateway common-gateway-to-all-spoke-peer {
            ike-policy common-cert-policy;
            dynamic {
                hostname example.net;
                ike-user-type group-ike-id;
            }
            external-interface fe-0/0/2;
        }
    }
}
```

In this example, `example.net` is the common part of the hostname identification used for all spokes. All X.509 certificates on the spokes must contain a hostname identity in the alternate subject field with `example.net` in the right-most part. All spokes must use the hostname identity in their IKE ID payload.

The following example configures a group IKE ID with wildcards to match the values `sales` in the organizational unit and `example` in the organization subject fields of the certificate.

```
[edit]
security {
    ike {
        policy common-cert-policy {
            proposals common-ike-proposal;
            certificate {
                local-certificate hub-local-certificate;
            }
        }
        gateway common-gateway-to-all-spoke-peer {
            ike-policy common-cert-policy;
            dynamic {
                distinguished-name {
                    wildcard ou=sales,o=example;
                }
                ike-user-type group-ike-id;
            }
            external-interface fe-0/0/2;
        }
    }
}
```

In this example, the fields `ou=sales,o=example` are the common part of the subject field in the certificates expected from the spokes. During IKE negotiation, if a spoke presents a certificate with the subject fields `cn=alice,ou=sales,o=example` in its certificate, authentication succeeds and the tunnel is established. If a spoke presents a certificate with the subject fields `cn=thomas,ou=engineer,o=example` in its certificate, the certificate is rejected by the hub as the organization unit should be `sales`.

Excluding a Spoke Connection

To exclude a particular spoke from connecting to the hub, the certificate for that spoke must be revoked. The hub needs to retrieve the latest certificate revocation list (CRL) from the CA that contains the serial number of the revoked certificate. The hub will then refuse a VPN connection from the revoked spoke. Until the latest CRL is available in the hub, the hub might continue to establish a tunnel from the revoked spoke. For more information, see "Understanding Online Certificate Status Protocol and Certificate Revocation Lists" on page 67 and "Understanding Certificate Authority Profiles" on page 46.

SEE ALSO

IPsec VPN with Autokey IKE Configuration Overview | **190**

AutoVPN Configuration Overview

The following steps describe the basic tasks for configuring AutoVPN on hub and spoke devices. The AutoVPN hub is configured *once* for all current and new spokes.

To configure the AutoVPN hub:

1. Enroll a CA certificate and the local certificate in the device.
2. Create a secure tunnel (st0) interface and configure it in point-to-multipoint mode.
3. Configure a single IKE policy.
4. Configure an IKE gateway with a group IKE ID that is common to all spokes.
5. Configure a single IPsec policy and VPN.
6. Configure a dynamic routing protocol.

To configure an SRX Series AutoVPN spoke device:

1. Enroll a CA certificate and the local certificate in the device.
2. Create an st0 interface and configure it in point-to-multipoint mode.
3. Configure an IKE policy to match the IKE policy configured on the hub.
4. Configure an IKE gateway with an ID to match the group IKE ID configured on the hub.
5. Configure an IPsec policy to match the IPsec policy configured on the hub.
6. Configure a dynamic routing protocol.

SEE ALSO

Understanding Traffic Selectors in Route-Based VPNs | 524

Example: Configuring Basic AutoVPN with iBGP

IN THIS SECTION

This example shows how to configure an AutoVPN hub to act as a single termination point, and then configure two spokes to act as tunnels to remote sites. This example configures iBGP to forward packets through the VPN tunnels.

Requirements

This example uses the following hardware and software components:

- Three supported SRX Series Firewalls as AutoVPN hub and spokes
- Junos OS Release 12.1X44-D10 and later that support AutoVPN

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels. For more information about specific requirements for a dynamic routing protocol, see the Routing Protocols Overview.

Overview

IN THIS SECTION

This example shows the configuration of an AutoVPN hub and the subsequent configurations of two spokes.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). The certificates for the spokes contain the organizational unit (OU) value “SLT” in the subject field; the hub is configured with a group IKE ID to match the value “SLT” in the OU field.

The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and all spokes must have the same values. Table 91 on page 1032 shows the options used in this example.

Table 91: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	2
Authentication algorithm	SHA-1
Encryption algorithm	AES 128 CBC
IKE policy:	
Mode	Main
IPsec proposal:	
Protocol	ESP
Authentication algorithm	HMAC MD5 96
Encryption algorithm	DES CBC
IPsec policy:	
Perfect Forward Secrecy (PFS) group	14

The same certificate authority (CA) is configured on all devices.

Junos OS only supports a single level of certificate hierarchy.

Table 92 on page 1033 shows the options configured on the hub and on all spokes.

Table 92: AutoVPN Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	10.1.1.1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string SLT in the organizational unit (OU) field	DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	ge-0/0/1.0	Spoke 1: fe-0/0/1.0 Spoke 2: ge-0/0/1.0
VPN:		
Bind interface	st0.0	st0.0
Establish tunnels	(not configured)	Immediately on configuration commit

Table 93 on page 1033 shows the configuration options that are different on each spoke.

Table 93: Comparison Between the Spoke Configurations

Option	Spoke 1	Spoke 2
st0.0 interface	10.10.10.2/24	10.10.10.3/24
Interface to internal network	(fe-0.0/4.0) 10.60.60.1/24	(fe-0.0/4.0) 10.70.70.1/24

Table 93: Comparison Between the Spoke Configurations *(Continued)*

Option	Spoke 1	Spoke 2
Interface to Internet	(fe-0/0/1.0) 10.2.2.1/30	(ge-0/0/1.0) 10.3.3.1/30

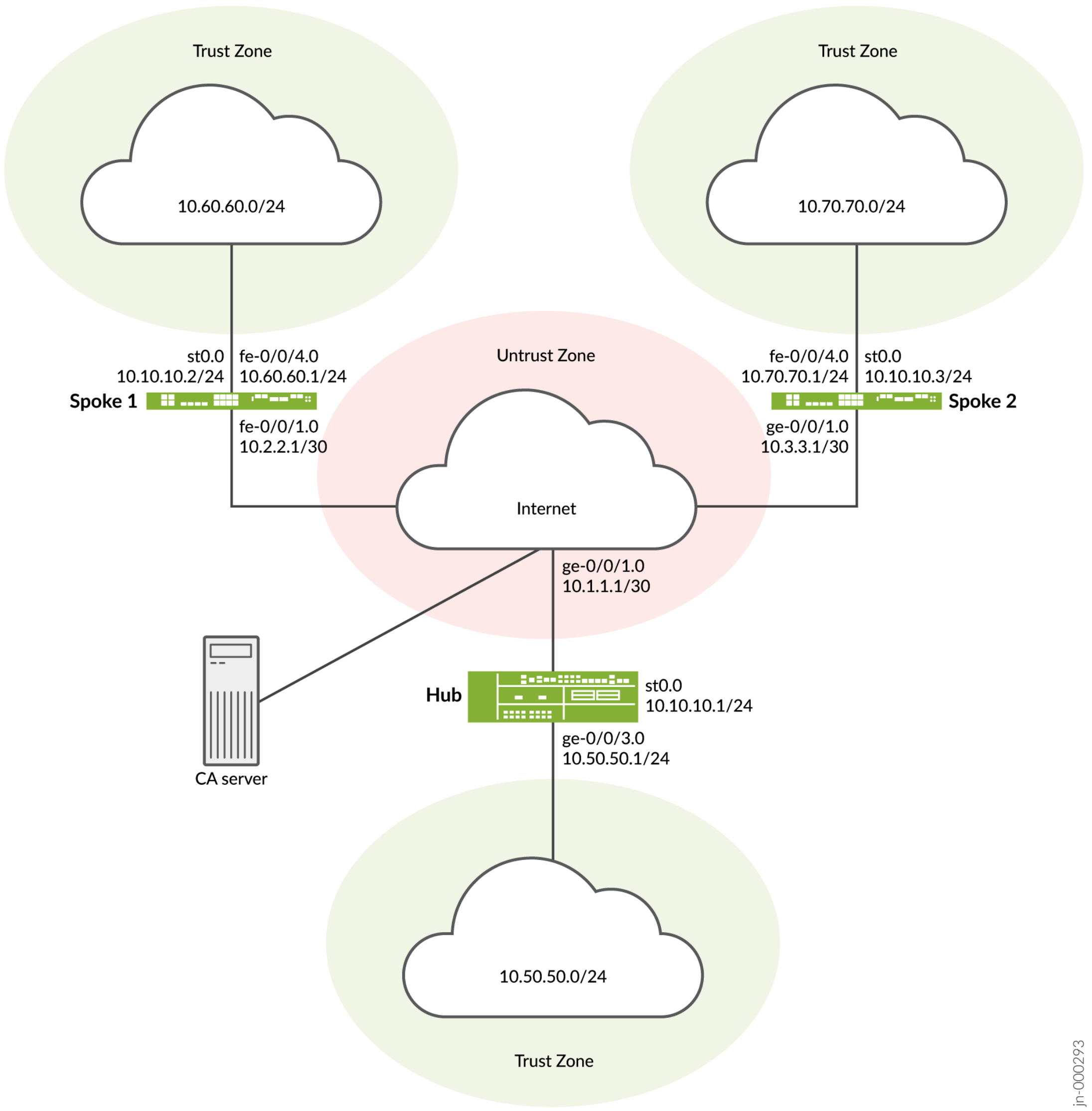
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Topology

Figure 66 on page 1035 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

Figure 66: Basic AutoVPN Deployment with iBGP



Configuration

IN THIS SECTION

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

   ```
   [edit]
   user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
   user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
   mscep.dll
   user@host# set security pki ca-profile ca-profile1 revocation-check disable
   user@host# commit
   ```

2. Enroll the CA certificate.

   ```
   user@host> request security pki ca-certificate enroll ca-profile ca-profile1
   ```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

   ```
   user@host> request security pki generate-key-pair certificate-id Local1
   ```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is SLT. The IKE configuration on the hub includes ou=SLT to identify the spoke.

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 2:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
```

```
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke2@example.net ip-address 10.3.3.1 subject
DC=example.net,CN=spoke2,OU=SLT,O=example,L=Tumkur,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40bb71d400000000258f
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Tumkur, Common name: spoke2, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
  Alternate subject: "spoke2@example.net", example.net, 10.3.3.1
  Validity:
    Not before: 11- 6-2012 10:02
    Not after: 11- 6-2013 10:12
  Public key algorithm: rsaEncryption(1024 bits)
   30:81:89:02:81:81:00:b6:2e:e2:da:e6:ac:57:e4:5d:ff:de:f6:89
   27:d6:3e:1b:4a:3f:b2:2d:b3:d3:61:ed:ed:6a:07:d9:8a:d2:24:03
   77:1a:fe:84:e1:12:8a:2d:63:6e:bf:02:6b:15:96:5a:4f:37:a0:46
   44:09:96:c0:fd:bb:ab:79:2c:5d:92:bd:31:f0:3b:29:51:ce:89:8e
   7c:2b:02:d0:14:5b:0a:a9:02:93:21:ea:f9:fc:4a:e7:08:bc:b1:6d
   7c:f8:3e:53:58:8e:f1:86:13:fe:78:b5:df:0b:8e:53:00:4a:46:11
   58:4a:38:e9:82:43:d8:25:47:7d:ef:18:f0:ef:a7:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
   http://ca-server1/CertEnroll/CASERVER1.crl
   file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
   1a:6d:77:ac:fd:94:68:ce:cf:8a:85:f0:39:fc:e0:6b:fd:fe:b8:66 (sha1)
   00:b1:32:5f:7b:24:9c:e5:02:e6:72:75:9e:a5:f4:77 (md5)
  Auto-re-enrollment:
   Status: Disabled
   Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/3 unit 0 family inet address 10.50.50.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.1/24
set policy-options policy-statement lan_nw from interface ge-0/0/3.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.10.10.1
set protocols bgp group ibgp export lan_nw
set protocols bgp group ibgp cluster 10.2.3.4
set protocols bgp group ibgp peer-as 65010
set policy-options policy-statement lan_nw from interface ge-0/0/3.0
set policy-options policy-statement lan_nw then accept
set policy-options policy-statement bgp_nh_self term 1 from protocol bgp
set policy-options policy-statement bgp_nh_self term 1 then next-hop self
set policy-options policy-statement bgp_nh_self term 1 then accept
set protocols bgp group ibgp export bgp_nh_self
set protocols bgp group ibgp allow 10.10.10.0/24
set routing-options static route 10.2.2.0/30 next-hop 10.1.1.2
set routing-options static route 10.3.3.0/30 next-hop 10.1.1.2
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway hub-to-spoke-gw ike-policy ike-policy1
set security ike gateway hub-to-spoke-gw dynamic distinguished-name wildcard OU=SLT
set security ike gateway hub-to-spoke-gw dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw local-identity distinguished-name
set security ike gateway hub-to-spoke-gw external-interface ge-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn hub-to-spoke-vpn bind-interface st0.0
set security ipsec vpn hub-to-spoke-vpn ike gateway hub-to-spoke-gw
set security ipsec vpn hub-to-spoke-vpn ike ipsec-policy vpn-policy1
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.1.1.1/30
user@host# set ge-0/0/3 unit 0 family inet address 10.50.50.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.1/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface ge-0/0/3.0
user@host# set policy-statement lan_nw then accept
user@host# set policy-statement bgp_nh_self term 1 from protocol bgp
user@host# set policy-statement bgp_nh_self term 1 then next-hop self
user@host# set policy-statement bgp_nh_self term 1 then accept
[edit protocols bgp]
user@host# set group ibgp type internal
user@host# set group ibgp local-address 10.10.10.1
user@host# set group ibgp export lan_nw
user@host# set group ibgp cluster 10.2.3.4
user@host# set group ibgp peer-as 65010
user@host# set group ibgp allow 10.10.10.0/24
user@host# set group ibgp export bgp_nh_self
[edit routing-options]
user@host# set static route 10.2.2.0/30 next-hop 10.1.1.2
user@host# set static route 10.3.3.0/30 next-hop 10.1.1.2
user@host# set autonomous-system 65010
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
```

```
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway hub-to-spoke-gw]
user@host# set ike-policy ike-policy1
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn hub-to-spoke-vpn]
user@host# set bind-interface st0.0
user@host# set ike gateway hub-to-spoke-gw
user@host# set ike ipsec-policy vpn-policy1
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/3.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.1/30;
        }
    }
}
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 10.50.50.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.1/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement bgp_nh_self {
    term 1 {
        from protocol bgp;
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement lan_nw {
    from interface ge-0/0/3.0;
    then accept;
}
[edit]
user@host# show protocols
bgp {
    group ibgp {
        type internal;
        local-address 10.10.10.1;
        export lan_nw;
        cluster 10.2.3.4;
        peer-as 65010;
        allow 10.10.10.0/24;
        export bgp_nh_self;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.2.2.0/30 next-hop 10.1.1.2;
    route 10.3.3.0/30 next-hop 10.1.1.2;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway hub-to-spoke-gw {
        ike-policy ike-policy1;
        dynamic {
            distinguished-name {
                wildcard OU=SLT;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
[edit]
user@host# show security ipsec
    proposal ipsec-proposal {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm des-cbc;
    }
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn hub-to-spoke-vpn {
        bind-interface st0.0;
        ike {
            gateway hub-to-spoke-gw;
            ipsec-policy vpn-policy1;
        }
    }
[edit]
```

```
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.0;
        ge-0/0/1.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/3.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces fe-0/0/1 unit 0 family inet address 10.2.2.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.60.60.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.2/24
set policy-options policy-statement lan_nw from interface fe-0/0/4.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.10.10.2
set protocols bgp group ibgp export lan_nw
set protocols bgp group ibgp neighbor 10.10.10.1
set routing-options static route 10.1.1.0/30 next-hop 10.2.2.2
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway spoke-to-hub-gw ike-policy ike-policy1
set security ike gateway spoke-to-hub-gw address 10.1.1.1
set security ike gateway spoke-to-hub-gw local-identity distinguished-name
set security ike gateway spoke-to-hub-gw remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw external-interface fe-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn spoke-to-hub bind-interface st0.0
set security ipsec vpn spoke-to-hub ike gateway spoke-to-hub-gw
set security ipsec vpn spoke-to-hub ike ipsec-policy vpn-policy1
set security ipsec vpn spoke-to-hub establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 1:

1. Configure interfaces.

   ```
   [edit interfaces]
   user@host# set fe-0/0/1 unit 0 family inet address 10.2.2.1/30
   user@host# set fe-0/0/4 unit 0 family inet address 10.60.60.1/24
   user@host# set st0 unit 0 multipoint
   user@host# set st0 unit 0 family inet address 10.10.10.2/24
   ```

2. Configure routing protocol.

   ```
   [edit policy-options]
   user@host# set policy-statement lan_nw from interface fe-0/0/4.0
   user@host# set policy-statement lan_nw then accept
   [edit protocols bgp]
   user@host# set group ibgp type internal
   user@host# set group ibgp local-address 10.10.10.2
   user@host# set group ibgp export lan_nw
   user@host# set group ibgp neighbor 10.10.10.1
   ```

```
[edit routing-options]
user@host# set static route 10.1.1.0/30 next-hop 10.2.2.2
user@host# set autonomous-system 10
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway spoke-to-hub-gw]
user@host# set ike-policy ike-policy1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub]
user@host# set bind-interface st0.0
user@host# set ike gateway spoke-to-hub-gw
user@host# set ike ipsec-policy vpn-policy1
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/1.0
user@host# set interfaces st0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
fe-0/0/1 {
    unit 0 {
        family inet {
            address 10.2.2.1/30;
        }
    }
}
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.60.60.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.2/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement lan_nw {
    from interface fe-0/0/4.0;
    then accept;
}
[edit]
user@host# show protocols
bgp {
    group ibgp {
        type internal;
        local-address 10.10.10.2;
        export lan_nw;
        neighbor 10.10.10.1;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.0/30 next-hop 10.2.2.2;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway spoke-to-hub-gw {
        ike-policy ike-policy1;
        address 10.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/1.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub {
        bind-interface st0.0;
        ike {
            gateway spoke-to-hub-gw;
            ipsec-policy vpn-policy1;
        }
        establish-tunnels immediately;
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-0/0/1.0;
        st0.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.70.70.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.3/24
set policy-options policy-statement lan_nw from interface fe-0/0/4.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.10.10.3
set protocols bgp group ibgp export lan_nw
set protocols bgp group ibgp neighbor 10.10.10.1
set routing-options static route 10.1.1.0/30 next-hop 10.3.3.2
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway spoke-to-hub-gw ike-policy ike-policy1
set security ike gateway spoke-to-hub-gw address 10.1.1.1
set security ike gateway spoke-to-hub-gw local-identity distinguished-name
set security ike gateway spoke-to-hub-gw remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw external-interface ge-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn spoke-to-hub bind-interface st0.0
set security ipsec vpn spoke-to-hub ike gateway spoke-to-hub-gw
set security ipsec vpn spoke-to-hub ike ipsec-policy vpn-policy1
set security ipsec vpn spoke-to-hub establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 10.70.70.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.3/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface fe-0/0/4.0
user@host# set policy-statement lan_nw then accept
[edit protocols bgp]
user@host# set group ibgp type internal
user@host# set group ibgp local-address 10.10.10.3
user@host# set group ibgp export lan_nw
user@host# set group ibgp neighbor 10.10.10.1
[edit routing-options]
user@host# set static route 10.1.1.0/30 next-hop 10.3.3.2
user@host# set autonomous-system 10
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway spoke-to-hub-gw]
user@host# set ike-policy ike-policy1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub]
user@host# set bind-interface st0.0
user@host# set ike gateway spoke-to-hub-gw
user@host# set ike ipsec-policy vpn-policy1
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.0
```

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.3.3.1/30;
        }
    }
}
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.70.70.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.3/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement lan_nw {
    from interface fe-0/0/4.0;
    then accept;
}
[edit]
user@host# show protocols
bgp {
    group ibgp {
        type internal;
        local-address 10.10.10.3;
        export lan_nw;
        neighbor 10.10.10.1;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.0/30 next-hop 10.3.3.2;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway spoke-to-hub-gw {
        ike-policy ike-policy1;
        address 10.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub {
        bind-interface st0.0;
        ike {
            gateway spoke-to-hub-gw;
            ipsec-policy vpn-policy1;
        }
        establish-tunnels immediately;
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
5480163 UP     a558717f387074ab  6d0135c5ecaed61d  Main           10.3.3.1
5480162 UP     7a63d16a5a723df1  c471f7ae166d3a34  Main           10.2.2.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the **security ipsec security-associations** command.

```
user@host> security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <268173400 ESP:des/ md5 9bf33bc7 3567/ unlim -   root 500   10.2.2.1
  >268173400 ESP:des/ md5 aae5196b 3567/ unlim -   root 500   10.2.2.1
  <268173401 ESP:des/ md5 69c24d81 622/ unlim  -   root 500   10.3.3.1
  >268173401 ESP:des/ md5 e3fe0231 622/ unlim  -   root 500   10.3.3.1
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway  interface   IPSec VPN name                    Flag     IKE-
ID                            XAUTH username
10.10.10.2        st0.0       hub-to-spoke-vpn                  Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
10.10.10.3        st0.0       hub-to-spoke-vpn                  Auto     C=IN, DC=example.net,
ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying BGP

Purpose

Verify that BGP references the IP addresses for the `st0` interfaces of the spokes.

Action

From operational mode, enter the **show bgp summary** command.

```
user@host> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Unconfigured peers: 2
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.10.10.2               10        116        119       0       0       50:25
1/1/1/0              0/0/0/0
10.10.10.3               10        114        114       0       0       50:04
1/1/1/0              0/0/0/0
```

Verifying Learned Routes

Purpose

Verify that routes to the spokes have been learned.

Action

From operational mode, enter the **show route 10.60.60.0** command.

```
user@host> show route 10.60.60.0
inet.0: 45 destinations, 45 routes (44 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.60.60.0/24      *[BGP/170] 00:50:57, localpref 100
                      AS path: I
                    > to 10.10.10.2 via st0.0
```

From operational mode, enter the **show route 10.70.70.0** command.

```
user@host> show route 10.70.70.0
inet.0: 45 destinations, 45 routes (44 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.70.70.0/24      *[BGP/170] 00:50:42, localpref 100
                      AS path: I
                    > to 10.10.10.3 via st0.0
```

SEE ALSO

Route-Based IPsec VPNs | 394

Routing Protocols Overview

Example: Configuring Basic AutoVPN with iBGP for IPv6 Traffic

IN THIS SECTION

This example shows how to configure an AutoVPN hub to act as a single termination point, and then configure two spokes to act as tunnels to remote sites. This example configures AutoVPN for IPv6 environment using iBGP to forward packets through the VPN tunnels.

Requirements

This example uses the following hardware and software components:

- Three supported SRX Series Firewalls as AutoVPN hub and spokes.
- Junos OS Release 18.1R1 and later releases.

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels. For more information about specific requirements for a dynamic routing protocol, see the Routing Protocols Overview.

Overview

IN THIS SECTION

This example shows the configuration of an AutoVPN hub and the subsequent configurations of two spokes .

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). The certificates for the spokes contain the organizational unit (OU) value “SLT” in the subject field; the hub is configured with a group IKE ID to match the value “SLT” in the OU field.

The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and all spokes must have the same values. Table 94 on page 1067 shows the options used in this example.

Table 94: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Configurations

Option	Value
IKE proposal:	

Table 94: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Configurations *(Continued)*

Option	Value
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	19
Authentication algorithm	SHA-384
Encryption algorithm	AES 256 CBC
IKE policy:	
Mode	Main
IPsec proposal:	
Protocol	ESP
Lifetime Seconds	3000
Encryption algorithm	AES 256 GCM
IPsec policy:	
Perfect Forward Secrecy (PFS) group	19

The same certificate authority (CA) is configured on all devices.

Junos OS only supports a single level of certificate hierarchy.

Table 95 on page 1069 shows the options configured on the hub and on all spokes.

Table 95: AutoVPN Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	2001:db8:2000::1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string `SLT` in the organizational unit (OU) field	DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	ge-0/0/0	Spoke 1: ge-0/0/0.0 Spoke 2: ge-0/0/0.0
VPN:		
Bind interface	st0.1	st0.1
Establish tunnels	(not configured)	establish-tunnels on-traffic

Table 96 on page 1069 shows the configuration options that are different on each spoke.

Table 96: Comparison Between the Spoke Configurations

Option	Spoke 1	Spoke 2
st0.0 interface	2001:db8:7000::2/64	2001:db8:7000::3/64
Interface to internal network	(ge-0/0/1.0) 2001:db8:4000::1/64	(ge-0/0/1.0) 2001:db8:6000::1/64
Interface to Internet	(ge-0/0/0.0) 2001:db8:3000::2/64	(ge-0/0/0.0) 2001:db8:5000::2/64

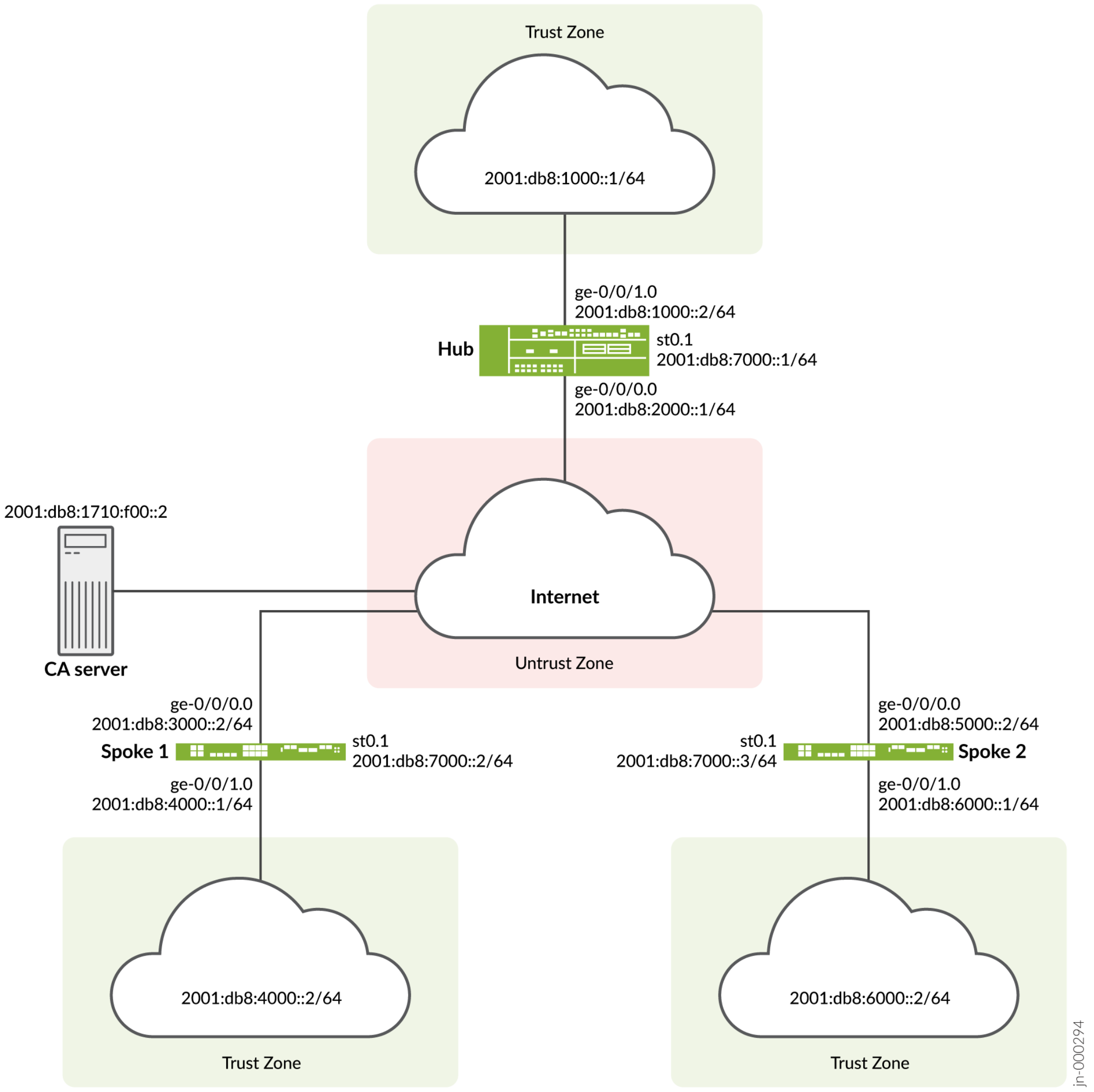
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Topology

Figure 67 on page 1070 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

Figure 67: Basic AutoVPN Deployment with iBGP



Configuration

IN THIS SECTION

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 2:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke2@example.net ip-address 10.3.3.1 subject
DC=example.net,CN=spoke2,OU=SLT,O=example,L=Tumkur,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40bb71d400000000258f
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
```

```
  Organization: example, Organizational unit: SLT, Country: IN, State: KA,
  Locality: Tumkur, Common name: spoke2, Domain component: example.net
Subject string:
  C=IN, DC=example.net, ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
Alternate subject: "spoke2@example.net", example.net, 10.3.3.1
Validity:
  Not before: 11- 6-2012 10:02
  Not after: 11- 6-2013 10:12
Public key algorithm: rsaEncryption(1024 bits)
  30:81:89:02:81:81:00:b6:2e:e2:da:e6:ac:57:e4:5d:ff:de:f6:89
  27:d6:3e:1b:4a:3f:b2:2d:b3:d3:61:ed:ed:6a:07:d9:8a:d2:24:03
  77:1a:fe:84:e1:12:8a:2d:63:6e:bf:02:6b:15:96:5a:4f:37:a0:46
  44:09:96:c0:fd:bb:ab:79:2c:5d:92:bd:31:f0:3b:29:51:ce:89:8e
  7c:2b:02:d0:14:5b:0a:a9:02:93:21:ea:f9:fc:4a:e7:08:bc:b1:6d
  7c:f8:3e:53:58:8e:f1:86:13:fe:78:b5:df:0b:8e:53:00:4a:46:11
  58:4a:38:e9:82:43:d8:25:47:7d:ef:18:f0:ef:a7:02:03:01:00:01
Signature algorithm: sha1WithRSAEncryption
Distribution CRL:
  http://ca-server1/CertEnroll/CASERVER1.crl
  file://\\ca-server1\CertEnroll\CASERVER1.crl
Fingerprint:
  1a:6d:77:ac:fd:94:68:ce:cf:8a:85:f0:39:fc:e0:6b:fd:fe:b8:66 (sha1)
  00:b1:32:5f:7b:24:9c:e5:02:e6:72:75:9e:a5:f4:77 (md5)
Auto-re-enrollment:
  Status: Disabled
  Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate HUB
set security ike gateway IKE_GWA_1 ike-policy IKE_POL
set security ike gateway IKE_GWA_1 dynamic distinguished-name wildcard OU=SLT
set security ike gateway IKE_GWA_1 dead-peer-detection always-send
set security ike gateway IKE_GWA_1 dead-peer-detection interval 10
set security ike gateway IKE_GWA_1 dead-peer-detection threshold 3
set security ike gateway IKE_GWA_1 local-identity distinguished-name
set security ike gateway IKE_GWA_1 external-interface ge-0/0/0
set security ike gateway IKE_GWA_1 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPNA_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPNA_1 ike gateway IKE_GWA_1
set security ipsec vpn IPSEC_VPNA_1 ike ipsec-policy IPSEC_POL
set security policies default-policy permit-all
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:2000::1/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1000::2/64
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::1/64
set routing-options rib inet6.0 static route 2001:db8:3000::/64 next-hop 2001:db8:2000::2
set routing-options rib inet6.0 static route 2001:db8:5000::/64 next-hop 2001:db8:2000::2
set routing-options autonomous-system 100
set routing-options forwarding-table export load_balance
set protocols bgp traceoptions file bgp
set protocols bgp traceoptions flag all
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 2001:db8:9000::1
set protocols bgp group ibgp export ibgp
set protocols bgp group ibgp cluster 10.1.3.4
set protocols bgp group ibgp peer-as 100
set protocols bgp group ibgp multipath
set protocols bgp group ibgp allow 2001:db8:9000::/64
set policy-options policy-statement ibgp from interface ge-0/0/1.0
set policy-options policy-statement ibgp then accept
set policy-options policy-statement load_balance then load-balance per-packet
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:2000::1/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:1000::2/64
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet6 address 2001:db8:7000::1/64
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement ibgp from interface ge-0/0/1.0
user@host# set policy-statement ibgp then accept
user@host# set policy-statement load_balance then load-balance per-packet
[edit protocols bgp]
user@host# set traceoptions file bgp
user@host# set traceoptions flag all
user@host# set group ibgp type internal
user@host# set group ibgp local-address 2001:db8:9000::1
user@host# set group ibgp export ibgp
```

```
user@host# set group ibgp cluster 10.1.3.4
user@host# set group ibgp peer-as 100
user@host# set group ibgp multipath
user@host# set group ibgp allow 2001:db8:9000::/64
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:3000::/64 next-hop 2001:db8:2000::2
user@host# set rib inet6.0 static route 2001:db8:5000::/64 next-hop 2001:db8:2000::2
user@host# set autonomous-system 100
user@host# set forwarding-table export load_balance
```

3. Configure Phase 1 options.

```
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike proposal ike-proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate HUB
[edit security ike gateway IKE_GWA_1]
user@host# set ike-policy IKE_POL
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
```

```
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPNA_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GWA_1
user@host# set ike ipsec-policy IPSEC_POL
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security` `policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:2000::1/64;
        }
    }
}
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:1000::2/64;
            }
        }
    }
    st0 {
        unit 1{
            multipoint;
            family inet6 {
                address 2001:db8:7000::1/64;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement ibgp {
    from interface ge-0/0/1.0;
    then accept;
}
policy-statement load_balance {
    then {
        load-balance per-packet;
    }
}
[edit]
user@host# show protocols
bgp {
    traceoptions {
        file bgp;
        flag all;
    }
    group ibgp {
        type internal;
        local-address 2001:db8:9000::1;
        export ibgp;
        cluster 10.1.3.4;
        peer-as 100;
        multipath;
        allow 2001:db8:9000::/64;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
        route route 2001:db8:3000::/64 next-hop 2001:db8:2000::2;
        route 2001:db8:5000::/64 next-hop 2001:db8:2000::2;
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate HUB;
    }
}
gateway IKE_GWA_1 {
    ike-policy IKE_POL;
    dynamic {
        distinguished-name {
            wildcard OU=SLT;
        }
    }
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    external-interface ge-0/0/0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPNA_1 {
    bind-interface st0.1;
    ike {
        gateway IKE_GWA_1;
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter commit from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE1
set security ike gateway IKE_GW_SPOKE_1 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_1 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_1 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_1 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_1 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_1 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike gateway IKE_GW_SPOKE_1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_1 establish-tunnels on-traffic
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::2/64
set routing-options rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:3000::1
set routing-options autonomous-system 100
set protocols bgp traceoptions file bgp
set protocols bgp traceoptions flag all
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 2001:db8:9000::2
set protocols bgp group ibgp export ibgp
set protocols bgp group ibgp peer-as 100
set protocols bgp group ibgp neighbor 2001:db8:9000::1
set policy-options policy-statement ibgp from interface ge-0/0/1.0
set policy-options policy-statement ibgp then accept
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 1:

1. Configure interfaces.

   ```
   [edit interfaces]
   user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
   user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
   user@host# set st0 unit 1 family inet6 address 2001:db8:7000::2/64
   ```

2. Configure routing protocol.

   ```
   [edit policy-options]
   user@host# set policy-statement ibgp from interface ge-0/0/1.0
   ```

```
user@host# set policy-statement ibgp then accept
[edit protocols bgp]
user@host# set traceoptions file bgp
user@host# set traceoptions flag all
user@host# set group ibgp type internal
user@host# set group ibgp local-address 2001:db8:9000::2
user@host# set group ibgp export ibgp
user@host# set group ibgp peer-as 100
user@host# set group ibgp neighbor 2001:db8:9000::1
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:3000::1
user@host# set autonomous-system 100
```

3. Configure Phase 1 options.

```
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike proposal ike-proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE1
[edit security ike gateway IKE_GW_SPOKE_1]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPNA_SPOKE_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GWA_SPOKE_1
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels on-traffic
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
```

```
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:3000::2/64;
        }
    }
}
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:4000::1/64;
            }
        }
    }
    st0 {
        unit 1{
            family inet6 {
                address 2001:db8:7000::2/64;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement ibgp {
    from interface ge-0/0/1.0;
    then accept;
}
[edit]
user@host# show protocols
```

```
bgp {
    traceoptions {
        file bgp;
        flag all;
    }
    group ibgp {
        type internal;
        local-address 2001:db8:9000::2;
        export ibgp;
        peer-as 100;
        neighbor 2001:db8:9000::1;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
        route route  2001:db8:2000::/64 next-hop 2001:db8:3000::1;
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE1;
    }
}
gateway IKE_GWA_SPOKE1 {
    ike-policy IKE_POL;
    dynamic {
        distinguished-name {
            wildcard OU=SLT;
        }
    }
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    external-interface ge-0/0/0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPNA_SPOKE_1 {
    bind-interface st0.1;
    ike {
        gateway IKE_GWA_SPOKE_1;
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE2
set security ike gateway IKE_GW_SPOKE_2 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_2 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_2 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_2 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_2 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_2 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_2 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike gateway IKE_GW_SPOKE_2
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_2 establish-tunnels on-traffic
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::3/64
set routing-options rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
set routing-options autonomous-system 100
set protocols bgp traceoptions file bgp
set protocols bgp traceoptions flag all
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 2001:db8:9000::3
set protocols bgp group ibgp export ibgp
set protocols bgp group ibgp peer-as 100
set protocols bgp group ibgp neighbor 2001:db8:9000::1
set policy-options policy-statement ibgp from interface ge-0/0/1.0
set policy-options policy-statement ibgp then accept
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
user@host# set st0 unit 1 family inet6 address 2001:db8:7000::3/64
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement ibgp from interface ge-0/0/1.0
```

```
user@host# set policy-statement ibgp then accept
[edit protocols bgp]
user@host# set traceoptions file bgp
user@host# set traceoptions flag all
user@host# set group ibgp type internal
user@host# set group ibgp local-address 2001:db8:9000::3
user@host# set group ibgp export ibgp
user@host# set group ibgp peer-as 100
user@host# set group ibgp neighbor 2001:db8:9000::1
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
user@host# set autonomous-system 100
```

3. Configure Phase 1 options.

```
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike proposal ike-proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE2
[edit security ike gateway IKE_GW_SPOKE_2]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPNA_SPOKE_2]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GWA_SPOKE_2
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels on-traffic
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
```

```
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:5000::2/64;
        }
    }
}
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:6000::1/64;
            }
        }
    }
    st0 {
        unit 1{
            family inet6 {
                address 2001:db8:7000::3/64;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement ibgp {
    from interface ge-0/0/1.0;
    then accept;
}
[edit]
user@host# show protocols
```

```
bgp {
    traceoptions {
        file bgp;
        flag all;
    }
    group ibgp {
        type internal;
        local-address 2001:db8:9000::3;
        export ibgp;
        peer-as 100;
        neighbor 2001:db8:9000::1;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
        route route 2001:db8:2000::/64 next-hop 2001:db8:5000::1;
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE2;
    }
}
gateway IKE_GWA_SPOKE2 {
    ike-policy IKE_POL;
    dynamic {
        distinguished-name {
            wildcard OU=SLT;
        }
    }
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    external-interface ge-0/0/0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPNA_SPOKE_2 {
    bind-interface st0.1;
    ike {
        gateway IKE_GWA_SPOKE_2;
        ipsec-policy IPSEC_POL;
    }
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying IKE Status

Purpose

Verify the IKE status.

Action

From operational mode, enter the **show security ike sa** command.

```
user@host> show security ike sa
Index  State Initiator cookie          Responder cookie          Mode Remote Address

493333 UP    2001:db8:88b49d915e684c93 2001:db8:fe890b1cac8522b5 Main 2001:db8:3000::2

493334 UP    2001:db8:26e40244ad3d722d 2001:db8:68b4d9f94097d32e Main 2001:db8:5000::2
```

Meaning

The `show security ike sa` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

Verifying IPsec Status

Purpose

Verify the IPsec status.

Action

From operational mode, enter the **show security ipsec sa** command.

```
user@host> show security ipsec sa
Total active tunnels: 2
  ID        Algorithm      SPI  Life:sec/kb    Mon    lsys Port Gateway
  >67108885 ESP:aes-gcm-256/None fdef4dab 2918/ unlim - root 500  2001:db8:3000::2
  >67108885 ESP:aes-gcm-256/None e785dadc 2918/ unlim - root 500  2001:db8:3000::2
  >67108887 ESP:aes-gcm-256/None 34a787af 2971/ unlim - root 500  2001:db8:5000::2
  >67108887 ESP:aes-gcm-256/None cf57007f 2971/ unlim - root 500  2001:db8:5000::2
```

Meaning

The `show security ipsec sa` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway              interface  IPSec VPN name  Flag  IKE-
ID                                XAUTH username

2001:db8:9000::2              st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available
```

```
2001:db8:9000::3              st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available

2001:db8::5668:ad10:fcd8:163c st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available

2001:db8::5668:ad10:fcd8:18a1 st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying BGP

Purpose

Verify that BGP references the IP addresses for the `st0` interfaces of the spokes.

Action

From operational mode, enter the **show bgp summary** command.

```
user@host> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Unconfigured peers: 2
Table       Tot Paths  Act Paths  Suppressed History Damp State    Pending
inet6.0
            2          2          0          0            0        0
Peer                AS   InPkt     OutPkt  OutQ  Flaps Last Up/Dwn State
2001:db8:9000::2    100  4          4      0     0          32     Establ
  inet6.0: 1/1/1/0
2001:db8:9000::3    100  4          4      0     0          8      Establ
  inet6.0: 1/1/1/0
```

SEE ALSO

Example: Configuring a Route-Based VPN | 395

Routing Protocols Overview

Example: Configuring AutoVPN with iBGP and ECMP

IN THIS SECTION

This example shows how to configure two IPsec VPN tunnels between an AutoVPN hub and spoke. This example configures iBGP with equal-cost multipath (ECMP) to forward packets through the VPN tunnels.

Requirements

This example uses the following hardware and software components:

- Two supported SRX Series Firewalls as AutoVPN hub and spoke
- Junos OS Release 12.1X44-D10 and later that support AutoVPN

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

This example shows the configuration of an AutoVPN hub and a spoke with two IPsec VPN tunnels.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). Certificates are enrolled in the hub and in the spoke for each IPsec VPN tunnel. One of the certificates for the spoke contains the organizational unit (OU) value “SLT” in the distinguished name (DN); the hub is configured with a group IKE ID to match the value “SLT” in the OU field. The other certificate for the spoke contains the OU value “SBU” in the DN; the hub is configured with a group IKE ID to match the value “SBU” in the OU field.

The spoke establishes IPsec VPN connections to the hub, which allows it to access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and the spoke must have the same values.Table 97 on page 1105 shows the options used in this example.

Table 97: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke iBGP ECMP Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	2
Authentication algorithm	SHA-1
Encryption algorithm	AES 128 CBC
IKE policy:	
Mode	Main

Table 97: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke iBGP ECMP Configurations *(Continued)*

Option	Value
IPsec proposal:	
Protocol	ESP
Authentication algorithm	HMAC MD5 96
Encryption algorithm	DES CBC
IPsec policy:	
Perfect Forward Secrecy (PFS) group	14

The same certificate authority (CA) is configured on all devices.

Junos OS only supports a single level of certificate hierarchy.

Table 98 on page 1106 shows the options configured on the hub and on the spoke.

Table 98: AutoVPN iBGP ECMP Configuration for Hub and Spoke 1

Option	Hub	Spoke 1
IKE gateway:		
Remote IP address	hub-to-spoke-gw-1: Dynamic hub-to-spoke-gw-2: Dynamic	spoke-to-hub-gw-1: 10.1.1.1 spoke-to-hub-gw-2: 10.1.2.1
Remote IKE ID	hub-to-spoke-gw-1: DN on the spoke's certificate with the string `SLT` in the OU field hub-to-spoke-gw-2: DN on the spoke's certificate with the string `SBU` in the OU field	spoke-to-hub-gw-1: DN on the hub's certificate spoke-to-hub-gw-2: DN on the hub's certificate

Table 98: AutoVPN iBGP ECMP Configuration for Hub and Spoke 1 *(Continued)*

Option	Hub	Spoke 1
Local IKE ID	DN on the hub’s certificate	DN on the spoke’s certificate
External interface	hub-to-spoke-gw-1: ge-0/0/1.0 hub-to-spoke-gw-2: ge-0/0/2.0	spoke-to-hub-gw-1: fe-0/0/1.0 spoke-to-hub-gw-2: fe-0/0/2.0
VPN:		
Bind interface	hub-to-spoke-vpn-1: st0.0 hub-to-spoke-vpn-2: st0.1	spoke-to-hub-1: st0.0 spoke-to-hub-2: st0.1
Establish tunnels	(not configured)	Immediately on configuration commit

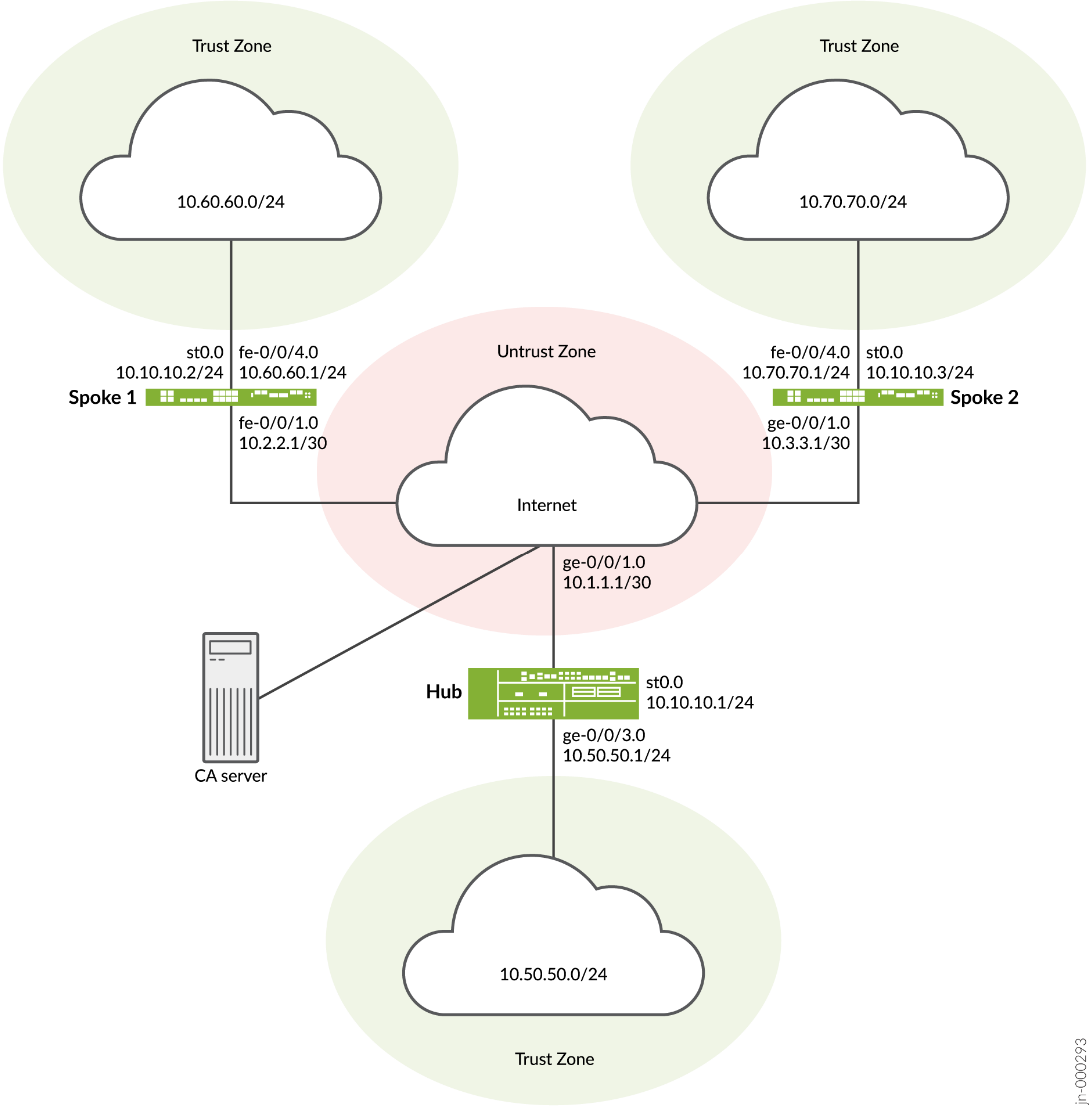
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Topology

Figure 68 on page 1108 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

Figure 68: AutoVPN Deployment with iBGP and ECMP



Configuration

IN THIS SECTION

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair for each certificate.

```
user@host> request security pki generate-key-pair certificate-id Local1
user@host> request security pki generate-key-pair certificate-id Local2
```

4. Enroll the local certificates.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local2 domain-name example.net email hub_backup@example.net ip-address 10.1.2.1 subject
```

```
DC=example.net,CN=hub_backup,OU=SBU,O=example,L=Bengaluru,ST=KA,C=IN challenge-password
<password>
```

5. Verify the local certificates.

```
user@host> show security pki local-certificate certificate-id Local1 detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

```
user@host> show security pki local-certificate certificate-id Local2 detail

Certificate identifier: Local2
  Certificate version: 3
  Serial number: 505efdf900000000259a
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SBU, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub_backup, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SBU, CN=hub_backup
  Alternate subject: "hub_backup@example.net", example.net, 10.1.2.1
  Validity:
    Not before: 11- 9-2012 10:55
    Not after: 11- 9-2013 11:05
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d5:44:08:96:f6:77:05:e6:91:50:8a:8a:2a
    4e:95:43:1e:88:ea:43:7c:c5:ac:88:d7:a0:8d:b5:d9:3f:41:db:db
    44:34:1f:56:a5:38:4b:b2:c5:85:f9:f1:bf:b2:7b:d4:b2:af:98:a0
    95:50:02:ad:f5:dd:4d:dc:67:85:dd:84:09:df:9c:68:a5:58:65:e7
    2c:72:cc:47:4b:d0:cc:4a:28:ca:09:db:ad:6e:5a:13:6c:e6:cc:f0
    29:ed:2b:2d:d1:38:38:bc:68:84:de:ae:86:39:c9:dd:06:d5:36:f0
    e6:2a:7b:46:4c:cd:a5:24:1c:e0:92:8d:ad:35:29:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    98:96:2f:ff:ca:af:33:ee:d7:4c:c8:4f:f7:71:53:c0:5d:5f:c5:59 (sha1)
    c9:87:e3:a4:5c:47:b5:aa:90:22:e3:06:b2:0b:e1:ea (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair for each certificate.

```
user@host> rrequest security pki generate-key-pair certificate-id Local1
user@host> request security pki generate-key-pair certificate-id Local2
```

4. Enroll the local certificates.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local2 domain-name example.net email spoke1_backup@example.net ip-address  10.3.3.1
subject DC=example.net,CN=spoke1_backup,OU=SBU,O=example,L=Mysore,ST=KA,C=IN challenge-
password <password>
```

5. Verify the local certificates.

```
user@host> show security pki local-certificate certificate-id Local1 detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started

user@host> show security pki local-certificate certificate-id Local2 detail

Certificate identifier: Local2
  Certificate version: 3
  Serial number: 506c3d0600000000259b
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SBU, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1_backup, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SBU, CN=spoke1_backup
  Alternate subject: "spoke1_backup@example.net", example.net, 10.3.3.1
  Validity:
    Not before: 11- 9-2012 11:09
    Not after: 11- 9-2013 11:19
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:a7:02:b5:e2:cd:79:24:f8:97:a3:8d:4d:27
    8c:2b:dd:f1:57:72:4d:2b:6d:d5:95:0d:9c:1b:5c:e2:a4:b0:84:2e
    31:82:3c:91:08:a2:58:b9:30:4c:5f:a3:6b:e6:2b:9c:b1:42:dd:1c
    cd:a2:7a:84:ea:7b:a6:b7:9a:13:33:c6:27:2b:79:2a:b1:0c:fe:08
    4c:a7:35:fc:da:4f:df:1f:cf:f4:ba:bc:5a:05:06:63:92:41:b4:f2
    54:00:3f:ef:ff:41:e6:ca:74:10:56:f7:2b:5f:d3:1a:33:7e:49:74
    1c:42:cf:c2:23:ea:4b:8f:50:2c:eb:1c:a6:37:89:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    d6:7f:52:a3:b6:f8:ae:cb:70:3f:a9:79:ea:8a:da:9e:ba:83:e4:5f (sha1)
    76:0b:72:73:cf:51:ee:58:81:2d:f7:b4:e2:5c:f4:5c (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT` for Local1 and `SBU` for Local2. The IKE configurations on the hub include `OU=SLT` and `OU=SBU` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/2 unit 0 family inet address 10.1.2.1/30
set interfaces ge-0/0/3 unit 0 family inet address 10.50.50.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.1/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 10.20.20.1/24
set policy-options policy-statement lan_nw from interface ge-0/0/3.0
set policy-options policy-statement lan_nw then accept
set policy-options policy-statement load_balance then load-balance per-packet
set protocols bgp group ibgp-1 type internal
set protocols bgp group ibgp-1 local-address 10.10.10.1
set protocols bgp group ibgp-1 export lan_nw
set protocols bgp group ibgp-1 cluster 10.2.3.4
set protocols bgp group ibgp-1 multipath
set protocols bgp group ibgp-1 allow 10.10.10.0/24
set protocols bgp group ibgp-2 type internal
set protocols bgp group ibgp-2 local-address 10.20.20.1
set protocols bgp group ibgp-2 export lan_nw
set protocols bgp group ibgp-2 cluster 10.2.3.5
set protocols bgp group ibgp-2 multipath
set protocols bgp group ibgp-2 allow 10.20.20.0/24
set routing-options static route 10.2.2.0/30 next-hop 10.1.1.2
set routing-options static route 10.3.3.0/30 next-hop 10.1.2.2
set routing-options autonomous-system 65010
set routing-options forwarding-table export load_balance
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy-1 mode main
set security ike policy ike-policy-1 proposals ike-proposal
set security ike policy ike-policy-1 certificate local-certificate Local1
set security ike policy ike-policy-2 mode main
set security ike policy ike-policy-2 proposals ike-proposal
set security ike policy ike-policy-2 certificate local-certificate Local2
set security ike gateway hub-to-spoke-gw-1 ike-policy ike-policy-1
set security ike gateway hub-to-spoke-gw-1 dynamic distinguished-name wildcard OU=SLT
set security ike gateway hub-to-spoke-gw-1 dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw-1 local-identity distinguished-name
set security ike gateway hub-to-spoke-gw-1 external-interface ge-0/0/1.0
set security ike gateway hub-to-spoke-gw-2 ike-policy ike-policy-2
set security ike gateway hub-to-spoke-gw-2 dynamic distinguished-name wildcard OU=SBU
set security ike gateway hub-to-spoke-gw-2 dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw-2 local-identity distinguished-name
set security ike gateway hub-to-spoke-gw-2 external-interface ge-0/0/2.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy proposals ipsec-proposal
set security ipsec vpn hub-to-spoke-vpn-1 bind-interface st0.0
set security ipsec vpn hub-to-spoke-vpn-1 ike gateway hub-to-spoke-gw-1
set security ipsec vpn hub-to-spoke-vpn-1 ike ipsec-policy vpn-policy
set security ipsec vpn hub-to-spoke-vpn-2 bind-interface st0.1
set security ipsec vpn hub-to-spoke-vpn-2 ike gateway hub-to-spoke-gw-2
set security ipsec vpn hub-to-spoke-vpn-2 ike ipsec-policy vpn-policy
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.1.1.1/30
user@host# set ge-0/0/2 unit 0 family inet address 10.1.2.1/30
user@host# set ge-0/0/3 unit 0 family inet address 10.50.50.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.1/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 10.20.20.1/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface ge-0/0/3.0
user@host# set policy-statement lan_nw then accept
user@host# set policy-statement load_balance then load-balance per-packet
```

```
[edit protocols bgp]
user@host# set group ibgp-1 type internal
user@host# set group ibgp-1 local-address 10.10.10.1
user@host# set group ibgp-1 export lan_nw
user@host# set group ibgp-1 cluster 10.2.3.4
user@host# set group ibgp-1 multipath
user@host# set group ibgp-1 allow 10.10.10.0/24
user@host# set group ibgp-2 type internal
user@host# set group ibgp-2 local-address 10.20.20.1
user@host# set group ibgp-2 export lan_nw
user@host# set group ibgp-2 cluster 10.2.3.5
user@host# set group ibgp-2 multipath
user@host# set group ibgp-2 allow 10.20.20.0/24
[edit routing-options]
user@host# set static route 10.2.2.0/30 next-hop 10.1.1.2
user@host# set static route 10.3.3.0/30 next-hop 10.1.2.2
user@host# set autonomous-system 65010
user@host# set forwarding-table export load_balance
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy-1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike policy ike-policy-2]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local2
[edit security ike gateway hub-to-spoke-gw-1]
user@host# set ike-policy ike-policy-1
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
[edit security ike gateway hub-to-spoke-gw-2]
user@host# set ike-policy ike-policy-2
```

```
user@host# set dynamic distinguished-name wildcard OU=SBU
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/2.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn hub-to-spoke-vpn-1]
user@host# set bind-interface st0.0
user@host# set ike gateway hub-to-spoke-gw-1
user@host# set ike ipsec-policy vpn-policy
[edit security ipsec vpn hub-to-spoke-vpn-2]
user@host# set bind-interface st0.1
user@host# set ike gateway hub-to-spoke-gw-2
user@host# set ike ipsec-policy vpn-policy
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.0
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces ge-0/0/2.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/3.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.1/30;
        }
    }
}
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 10.1.2.1/30;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 10.50.50.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.1/24;
            }
        }
        unit 1 {
            multipoint;
            family inet {
                address 10.20.20.1/24;
            }
        }
    }
[edit]
user@host# **show policy-options**
policy-statement lan_nw {
    from interface ge-0/0/3.0;
    then accept;
}
    policy-statement load_balance {
        then {
            load-balance per-packet;
        }
    }
[edit]
user@host# **show protocols**
bgp {
    group ibgp-1 {
        type internal;
        local-address 10.10.10.1;
        export lan_nw;
        cluster 10.2.3.4;
        multipath;
        allow 10.10.10.0/24;
    }
    group ibgp-2 {
        type internal;
        local-address 10.20.20.1;
        export lan_nw;
        cluster 10.2.3.5;
        multipath;
        allow 10.20.20.0/24;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.2.2.0/30 next-hop 10.1.1.2;
    route 10.3.3.0/30 next-hop 10.1.2.2;
    }
autonomous-system 65010;
    forwarding-table {
        export load_balance;
    }
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy-1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    policy ike-policy-2 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local2;
        }
    }
    gateway hub-to-spoke-gw-1 {
        ike-policy ike-policy-1;
        dynamic {
            distinguished-name {
                wildcard OU=SLT;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
    gateway hub-to-spoke-gw-2 {
        ike-policy ike-policy-2;
        dynamic {
            distinguished-name {
                wildcard OU=SBU;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/2.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn hub-to-spoke-vpn-1 {
        bind-interface st0.0;
        ike {
            gateway hub-to-spoke-gw-1;
            ipsec-policy vpn-policy;
        }
    }
    vpn hub-to-spoke-vpn-2 {
        bind-interface st0.1;
        ike {
            gateway hub-to-spoke-gw-2;
            ipsec-policy vpn-policy;
        }
    }
```

```
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.0;
        ge-0/0/1.0;
        ge-0/0/2.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/3.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces fe-0/0/1 unit 0 family inet address 10.2.2.1/30
set interfaces fe-0/0/2 unit 0 family inet address 10.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.60.60.1/24
set interfaces st0 unit 0 family inet address 10.10.10.2/24
set interfaces st0 unit 1 family inet address 10.20.20.2/24
set policy-options policy-statement lan_nw from interface fe-0/0/4.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp-1 type internal
set protocols bgp group ibgp-1 local-address 10.10.10.2
set protocols bgp group ibgp-1 export lan_nw
set protocols bgp group ibgp-1 neighbor 10.10.10.1
set protocols bgp group ibgp-2 type internal
set protocols bgp group ibgp-2 local-address 10.20.20.2
set protocols bgp group ibgp-2 export lan_nw
set protocols bgp group ibgp-2 neighbor 10.20.20.1
set routing-options static route 10.1.1.0/30 next-hop 10.2.2.2
set routing-options static route 10.1.2.0/30 next-hop 10.3.3.2
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy-1 mode main
set security ike policy ike-policy-1 proposals ike-proposal
set security ike policy ike-policy-1 certificate local-certificate Local1
set security ike policy ike-policy-2 mode main
set security ike policy ike-policy-2 proposals ike-proposal
set security ike policy ike-policy-2 certificate local-certificate Local2
set security ike gateway spoke-to-hub-gw-1 ike-policy ike-policy-1
set security ike gateway spoke-to-hub-gw-1 address 10.1.1.1
set security ike gateway spoke-to-hub-gw-1 local-identity distinguished-name
set security ike gateway spoke-to-hub-gw-1 remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw-1 external-interface fe-0/0/1.0
set security ike gateway spoke-to-hub-gw-2 ike-policy ike-policy-2
set security ike gateway spoke-to-hub-gw-2 address 10.1.2.1
set security ike gateway spoke-to-hub-gw-2 local-identity distinguished-name
set security ike gateway spoke-to-hub-gw-2 remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw-2 external-interface fe-0/0/2.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy proposals ipsec-proposal
set security ipsec vpn spoke-to-hub-1 bind-interface st0.0
set security ipsec vpn spoke-to-hub-1 ike gateway spoke-to-hub-gw-1
set security ipsec vpn spoke-to-hub-1 ike ipsec-policy vpn-policy
set security ipsec vpn spoke-to-hub-1 establish-tunnels immediately
set security ipsec vpn spoke-to-hub-2 bind-interface st0.1
set security ipsec vpn spoke-to-hub-2 ike gateway spoke-to-hub-gw-2
set security ipsec vpn spoke-to-hub-2 ike ipsec-policy vpn-policy
set security ipsec vpn spoke-to-hub-2 establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces fe-0/0/2.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set fe-0/0/1 unit 0 family inet address 10.2.2.1/30
user@host# set fe-0/0/2 unit 0 family inet address 10.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 10.60.60.1/24
user@host# set st0 unit 0 family inet address 10.10.10.2/24
user@host# set st0 unit 1 family inet address 10.20.20.2/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface fe-0/0/4.0
user@host# set policy-statement lan_nw then accept
[edit protocols bgp]
user@host# set group ibgp-1 type internal
user@host# set group ibgp-1 local-address 10.10.10.2
user@host# set group ibgp-1 export lan_nw
user@host# set group ibgp-1 neighbor 10.10.10.1
user@host# set group ibgp-2 type internal
user@host# set group ibgp-2 local-address 10.20.20.2
user@host# set group ibgp-2 export lan_nw
user@host# set group ibgp-2 neighbor 10.20.20.1
[edit routing-options]
user@host# set static route 10.1.1.0/30 next-hop 10.2.2.2
user@host# set static route 10.1.2.0/30 next-hop 10.3.3.2
user@host# set autonomous-system 65010
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy-1]
user@host# set mode main
user@host# set proposals ike-proposal
```

```
user@host# set certificate local-certificate Local1
[edit security ike policy ike-policy-2]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local2
[edit security ike gateway spoke-to-hub-gw-1]
user@host# set ike-policy ike-policy-1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/1.0
[edit security ike gateway spoke-to-hub-gw-2]
user@host# set ike-policy ike-policy-2
user@host# set address 10.1.2.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/2.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub-1]
user@host# set bind-interface st0.0
user@host# set ike gateway spoke-to-hub-gw-1
user@host# set ike ipsec-policy vpn-policy
user@host# set establish-tunnels immediately
[edit security ipsec vpn spoke-to-hub-2]
user@host# set bind-interface st0.1
user@host# set ike gateway spoke-to-hub-gw-2
user@host# set ike ipsec-policy vpn-policy
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/1.0
user@host# set interfaces st0.0
user@host# set interfaces fe-0/0/2.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
fe-0/0/1 {
    unit 0 {
        family inet {
            address 10.2.2.1/30;
        }
    }
}
    fe-0/0/2 {
        unit 0 {
            family inet {
                address 10.3.3.1/30;
            }
        }
    }
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.60.60.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            family inet {
                address 10.10.10.2/24;
            }
        }
        unit 1 {
            family inet {
                address 10.20.20.2/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement lan_nw {
    from interface fe-0/0/4.0;
    then accept;
}
[edit]
user@host# show protocols
bgp {
    group ibgp-1 {
        type internal;
        local-address 10.10.10.2;
        export lan_nw;
        neighbor 10.10.10.1;
    }
    group ibgp-2 {
        type internal;
        local-address 10.20.20.2;
        export lan_nw;
        neighbor 10.20.20.1;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.0/30 next-hop 10.2.2.2;
    route 10.1.2.0/30 next-hop 10.3.3.2;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy-1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    policy ike-policy-2 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local2;
        }
    }
    gateway spoke-to-hub-gw-1 {
        ike-policy ike-policy-1;
        address 1o.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/1.0;
    }
    gateway spoke-to-hub-gw-2 {
        ike-policy ike-policy-2;
        address 1o.1.2.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/2.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub-1 {
        bind-interface st0.0;
        ike {
            gateway spoke-to-hub-gw-1;
            ipsec-policy vpn-policy;
        }
        establish-tunnels immediately;
    }
    vpn spoke-to-hub-2 {
        bind-interface st0.1;
        ike {
            gateway spoke-to-hub-gw-2;
            ipsec-policy vpn-policy;
        }
        establish-tunnels immediately;
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-0/0/1.0;
        st0.0;
        fe-0/0/2.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
3733049 UP     bc9686796c2e52e9  1fbe46eee168f24e  Main           10.2.2.1
3733048 UP     a88db7ed23ec5f6b  c88b81dff52617a5  Main           10.3.3.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spoke.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the **security ipsec security-associations** command.

```
user@host> security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <268173315 ESP:des/ md5 93cfb417 1152/ unlim -   root 500   10.2.2.1
  >268173315 ESP:des/ md5 101de6f7 1152/ unlim -   root 500   10.2.2.1
  <268173313 ESP:des/ md5 272e29c0 1320/ unlim -   root 500   10.3.3.1
  >268173313 ESP:des/ md5 a3bf8fad 1320/ unlim -   root 500   10.3.3.1
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spoke.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway  interface   IPSec VPN name                  Flag     IKE-
ID                            XAUTH username
10.10.10.2        st0.0       hub-to-spoke-vpn-1              Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
```

```
10.20.20.2       st0.1      hub-to-spoke-vpn-2                Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SBU, CN=spoke1_backup
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying BGP

Purpose

Verify that BGP references the IP addresses for the `st0` interfaces of the spoke.

Action

From operational mode, enter the **show bgp summary** command.

```
user@host> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Unconfigured peers: 2
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.10.10.2           65010       4819       4820       0       2 1d 12:15:14
1/1/1/0             0/0/0/0
10.20.20.2           65010       4926       4928       0       0 1d 13:03:03
1/1/1/0             0/0/0/0
```

Verifying Learned Routes

Purpose

Verify that routes to the spoke have been learned.

Action

From operational mode, enter the **show route 10.60.60.0 detail** command.

```
user@host> show route 10.60.60.0 detail
inet.0: 47 destinations, 48 routes (46 active, 0 holddown, 1 hidden)
10.60.60.0/24 (2 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0x167407c
                Next-hop reference count: 3
                Source: 10.10.10.2
                Next hop type: Router
                Next hop: 10.10.10.2 via st0.0
                Next hop type: Router
                Next hop: 10.20.20.2 via st0.1, selected
                Protocol next hop: 10.10.10.2
                Indirect next hop: 15c8000 262142
                Protocol next hop: 10.20.20.2
                Indirect next hop: 15c80e8 262143
                State: <Act Int Ext>
                Local AS:    65010 Peer AS:    65010
                Age: 1d 12:16:25     Metric2: 0
                Task: BGP_10.10.10.10.2+53120
                Announcement bits (2): 0-KRT 3-Resolve tree 1
                AS path: I
                Accepted Multipath
                Localpref: 100
                Router ID: 10.207.36.182
         BGP    Preference: 170/-101
                Next hop type: Indirect
                Address: 0x15b8ac0
                Next-hop reference count: 1
                Source: 10.20.20.2
                Next hop type: Router
                Next hop: 10.20.20.2 via st0.1, selected
                Protocol next hop: 10.20.20.2
                Indirect next hop: 15c80e8 262143
                State: <NotBest Int Ext>
                Inactive reason: Not Best in its group - Update source
                Local AS:    65010 Peer AS:    65010
                Age: 1d 13:04:14     Metric2: 0
```

```
                  Task: BGP_10.20.20.20.2+50733
                  AS path: I
                  Accepted MultipathContrib
                  Localpref: 100
                  Router ID: 10.207.36.182
```

Verifying Route Installation in Forwarding Table

Purpose

Verify that routes to the spoke have been installed in the forwarding table.

Action

From operational mode, enter the **show route forwarding-table matching 10.60.60.0** command.

```
user@host> show route forwarding-table matching 60.60.60.0
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
10.60.60.0/24      user     0                    ulst 262144     1
                                                 indr 262142     2
                              10.10.10.2         ucst   572     3 st0.0
                                                 indr 262143     2
                              10.20.20.2         ucst   573     3 st0.1
```

SEE ALSO

Route-Based IPsec VPNs | 394

Example: Configuring AutoVPN with iBGP and Active-Backup Tunnels

IN THIS SECTION

This example shows how to configure active and backup IPsec VPN tunnels between an AutoVPN hub and spoke. This example configures iBGP to forward traffic through the VPN tunnels.

Requirements

This example uses the following hardware and software components:

- Two supported SRX Series Firewalls as AutoVPN hub and spoke
- Junos OS Release 12.1X44-D10 and later that support AutoVPN

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

This example shows the configuration of an AutoVPN hub and a spoke with two IPsec VPN tunnels.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). Certificates are enrolled in the hub and in the spoke for each IPsec VPN tunnel. One of the certificates for the spoke contains the organizational unit (OU) value “SLT” in the distinguished name (DN); the hub is configured with a group IKE ID to match the value “SLT” in the OU field. The other certificate for the spoke contains the OU value “SBU” in the DN; the hub is configured with a group IKE ID to match the value “SBU” in the OU field.

The spoke establishes IPsec VPN connections to the hub, which allows it to access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and the spoke must have the same values. Table 99 on page 1139 shows the options used in this example.

Table 99: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke iBGP Active-Backup Tunnel Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	2
Authentication algorithm	SHA-1
Encryption algorithm	AES 128 CBC
IKE policy:	
Mode	Main
IPsec proposal:	
Protocol	ESP
Authentication algorithm	HMAC MD5 96
Encryption algorithm	DES CBC
IPsec policy:	
Perfect Forward Secrecy (PFS) group	14

The same certificate authority (CA) is configured on all devices.

Junos OS only supports a single level of certificate hierarchy.

Table 100 on page 1140 shows the options configured on the hub and on the spoke.

Table 100: AutoVPN IBGP Active-Backup Tunnel Configuration for Hub and Spoke 1

Option	Hub	Spoke 1
IKE gateway:		
Remote IP address	hub-to-spoke-gw-1: Dynamic hub-to-spoke-gw-2: Dynamic	spoke-to-hub-gw-1: 10.1.1.1 spoke-to-hub-gw-2: 10.1.2.1
Remote IKE ID	hub-to-spoke-gw-1: DN on the spoke's certificate with the string SLT in the OU field hub-to-spoke-gw-2: DN on the spoke's certificate with the string SBU in the OU field	spoke-to-hub-gw-1: DN on the hub's certificate spoke-to-hub-gw-2: DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	hub-to-spoke-gw-1: ge-0/0/1.0 hub-to-spoke-gw-2: ge-0/0/2.0	spoke-to-hub-gw-1: fe-0/0/1.0 spoke-to-hub-gw-2: fe-0/0/2.0
VPN:		
Bind interface	hub-to-spoke-vpn-1: st0.0 hub-to-spoke-vpn-2: st0.1	spoke-to-hub-1: st0.0 spoke-to-hub-2: st0.1
VPN monitor	hub-to-spoke-vpn-1: ge-0/0/1.0 (source interface) hub-to-spoke-vpn-2: ge-0/0/2.0 (source interface)	spoke-to-hub-1: 10.1.1.1 (destination IP) spoke-to-hub-2: 10.1.2.1 (destination IP)
Establish tunnels	(not configured)	Immediately on configuration commit

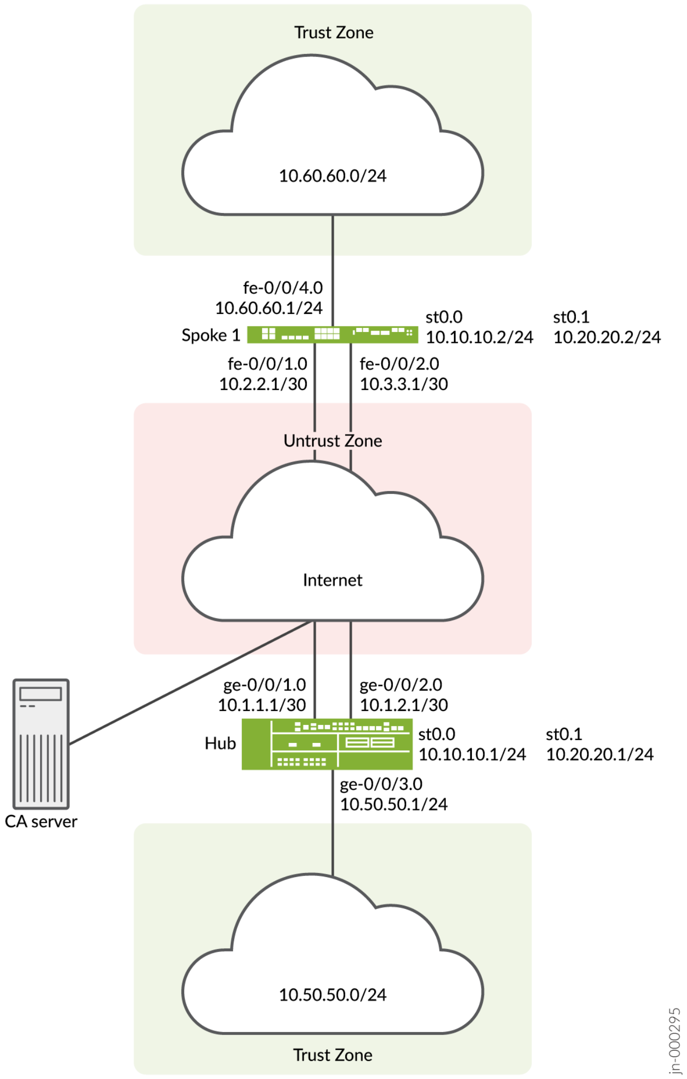
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Topology

Figure 69 on page 1141 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

Figure 69: AutoVPN Deployment with iBGP and Active-Backup Tunnels



In this example, two IPsec VPN tunnels are established between the hub and spoke 1. Routing information is exchanged through iBGP sessions in each tunnel. The longest prefix match for the route to 10.60.60.0/24 is through the st0.0 interface on the hub. Thus, the primary tunnel for the route is through the st0.0 interfaces on the hub and spoke 1. The default route is through the backup tunnel on the st0.1 interfaces on the hub and spoke 1.

VPN monitoring checks the status of the tunnels. If there is a problem with the primary tunnel (for example, the remote tunnel gateway is not reachable), the tunnel status changes to down and data destined for 10.60.60.0/24 is rerouted through the backup tunnel.

Configuration

IN THIS SECTION

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair for each certificate.

```
user@host> request security pki generate-key-pair certificate-id Local1
user@host> request security pki generate-key-pair certificate-id Local2
```

4. Enroll the local certificates.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local2 domain-name example.net email hub_backup@example.net ip-address 10.1.2.1 subject
DC=example.net,CN=hub_backup,OU=SBU,O=example,L=Bengaluru,ST=KA,C=IN challenge-password
<password>
```

5. Verify the local certificates.

```
user@host> show security pki local-certificate certificate-id Local1 detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

```
user@host> show security pki local-certificate certificate-id Local2 detail

Certificate identifier: Local2
  Certificate version: 3
  Serial number: 505efdf900000000259a
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SBU, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub_backup, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SBU, CN=hub_backup
  Alternate subject: "hub_backup@example.net", example.net, 10.1.2.1
  Validity:
    Not before: 11- 9-2012 10:55
    Not after: 11- 9-2013 11:05
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d5:44:08:96:f6:77:05:e6:91:50:8a:8a:2a
    4e:95:43:1e:88:ea:43:7c:c5:ac:88:d7:a0:8d:b5:d9:3f:41:db:db
    44:34:1f:56:a5:38:4b:b2:c5:85:f9:f1:bf:b2:7b:d4:b2:af:98:a0
    95:50:02:ad:f5:dd:4d:dc:67:85:dd:84:09:df:9c:68:a5:58:65:e7
    2c:72:cc:47:4b:d0:cc:4a:28:ca:09:db:ad:6e:5a:13:6c:e6:cc:f0
    29:ed:2b:2d:d1:38:38:bc:68:84:de:ae:86:39:c9:dd:06:d5:36:f0
    e6:2a:7b:46:4c:cd:a5:24:1c:e0:92:8d:ad:35:29:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    98:96:2f:ff:ca:af:33:ee:d7:4c:c8:4f:f7:71:53:c0:5d:5f:c5:59 (sha1)
    c9:87:e3:a4:5c:47:b5:aa:90:22:e3:06:b2:0b:e1:ea (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair for each certificate.

```
user@host> rrequest security pki generate-key-pair certificate-id Local1
user@host> request security pki generate-key-pair certificate-id Local2
```

4. Enroll the local certificates.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local2 domain-name example.net email spoke1_backup@example.net ip-address  10.3.3.1
```

```
subject DC=example.net,CN=spoke1_backup,OU=SBU,O=example,L=Mysore,ST=KA,C=IN challenge-
password <password>
```

5. Verify the local certificates.

```
user@host> show security pki local-certificate certificate-id Local1 detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started

user@host> show security pki local-certificate certificate-id Local2 detail
```

```
Certificate identifier: Local2
  Certificate version: 3
  Serial number: 506c3d0600000000259b
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SBU, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1_backup, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SBU, CN=spoke1_backup
  Alternate subject: "spoke1_backup@example.net", example.net, 10.3.3.1
  Validity:
    Not before: 11- 9-2012 11:09
    Not after: 11- 9-2013 11:19
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:a7:02:b5:e2:cd:79:24:f8:97:a3:8d:4d:27
    8c:2b:dd:f1:57:72:4d:2b:6d:d5:95:0d:9c:1b:5c:e2:a4:b0:84:2e
    31:82:3c:91:08:a2:58:b9:30:4c:5f:a3:6b:e6:2b:9c:b1:42:dd:1c
    cd:a2:7a:84:ea:7b:a6:b7:9a:13:33:c6:27:2b:79:2a:b1:0c:fe:08
    4c:a7:35:fc:da:4f:df:1f:cf:f4:ba:bc:5a:05:06:63:92:41:b4:f2
    54:00:3f:ef:ff:41:e6:ca:74:10:56:f7:2b:5f:d3:1a:33:7e:49:74
    1c:42:cf:c2:23:ea:4b:8f:50:2c:eb:1c:a6:37:89:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    d6:7f:52:a3:b6:f8:ae:cb:70:3f:a9:79:ea:8a:da:9e:ba:83:e4:5f (sha1)
    76:0b:72:73:cf:51:ee:58:81:2d:f7:b4:e2:5c:f4:5c (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is SLT for Local1 and SBU for Local2. The IKE configurations on the hub include OU=SLT and OU=SBU to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/2 unit 0 family inet address 10.1.2.1/30
set interfaces ge-0/0/3 unit 0 family inet address 10.50.50.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.1/24
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet address 10.20.20.1/24
set policy-options policy-statement lan_nw from interface ge-0/0/3.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp-1 type internal
set protocols bgp group ibgp-1 local-address 10.10.10.1
set protocols bgp group ibgp-1 export lan_nw
set protocols bgp group ibgp-1 cluster 10.2.3.4
set protocols bgp group ibgp-1 allow 10.10.10.0/24
set protocols bgp group ibgp-2 type internal
set protocols bgp group ibgp-2 local-address 10.20.20.1
set protocols bgp group ibgp-2 export lan_nw
set protocols bgp group ibgp-2 cluster 10.2.3.5
set protocols bgp group ibgp-2 allow 10.20.20.0/24
set routing-options static route 10.2.2.0/30 next-hop 10.1.1.2
set routing-options static route 10.3.3.0/30 next-hop 10.1.2.2
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy-1 mode main
set security ike policy ike-policy-1 proposals ike-proposal
set security ike policy ike-policy-1 certificate local-certificate Local1
set security ike policy ike-policy-2 mode main
set security ike policy ike-policy-2 proposals ike-proposal
set security ike policy ike-policy-2 certificate local-certificate Local2
set security ike gateway hub-to-spoke-gw-1 ike-policy ike-policy-1
set security ike gateway hub-to-spoke-gw-1 dynamic distinguished-name wildcard OU=SLT
set security ike gateway hub-to-spoke-gw-1 dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw-1 local-identity distinguished-name
set security ike gateway hub-to-spoke-gw-1 external-interface ge-0/0/1.0
set security ike gateway hub-to-spoke-gw-2 ike-policy ike-policy-2
set security ike gateway hub-to-spoke-gw-2 dynamic distinguished-name wildcard OU=SBU
set security ike gateway hub-to-spoke-gw-2 dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw-2 local-identity distinguished-name
set security ike gateway hub-to-spoke-gw-2 external-interface ge-0/0/2.0
set security ipsec vpn-monitor-options interval 5
set security ipsec vpn-monitor-options threshold 2
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy proposals ipsec-proposal
set security ipsec vpn hub-to-spoke-vpn-1 bind-interface st0.0
set security ipsec vpn hub-to-spoke-vpn-1 vpn-monitor source-interface ge-0/0/1.0
set security ipsec vpn hub-to-spoke-vpn-1 ike gateway hub-to-spoke-gw-1
set security ipsec vpn hub-to-spoke-vpn-1 ike ipsec-policy vpn-policy
set security ipsec vpn hub-to-spoke-vpn-2 bind-interface st0.1
set security ipsec vpn hub-to-spoke-vpn-2 vpn-monitor source-interface ge-0/0/2.0
set security ipsec vpn hub-to-spoke-vpn-2 ike gateway hub-to-spoke-gw-2
set security ipsec vpn hub-to-spoke-vpn-2 ike ipsec-policy vpn-policy
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.1.1.1/30
user@host# set ge-0/0/2 unit 0 family inet address 10.1.2.1/30
user@host# set ge-0/0/3 unit 0 family inet address 10.50.50.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.1/24
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet address 10.20.20.1/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement lan_nw from interface ge-0/0/3.0
user@host# set policy-statement lan_nw then accept
[edit protocols bgp]
user@host# set group ibgp-1 type internal
user@host# set group ibgp-1 local-address 10.10.10.1
user@host# set group ibgp-1 export lan_nw
user@host# set group ibgp-1 cluster 10.2.3.4
user@host# set group ibgp-1 allow 10.10.10.0/24
user@host# set group ibgp-2 type internal
user@host# set group ibgp-2 local-address 10.20.20.1
user@host# set group ibgp-2 export lan_nw
user@host# set group ibgp-2 cluster 10.2.3.5
user@host# set group ibgp-2 allow 10.20.20.0/24
[edit routing-options]
user@host# set static route 10.2.2.0/30 next-hop 10.1.1.2
user@host# set static route 10.3.3.0/30 next-hop 10.1.2.2
user@host# set autonomous-system 65010
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
```

```
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy-1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike policy ike-policy-2]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local2
[edit security ike gateway hub-to-spoke-gw-1]
user@host# set ike-policy ike-policy-1
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
[edit security ike gateway hub-to-spoke-gw-2]
user@host# set ike-policy ike-policy-2
user@host# set dynamic distinguished-name wildcard OU=SBU
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/2.0
```

4. Configure Phase 2 options.

```
[edit security ipsec vpn-monitor]
user@host# set options interval 5
user@host# set options threshold 2
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn hub-to-spoke-vpn-1]
user@host# set bind-interface st0.0
user@host# set vpn-monitor source-interface ge-0/0/1.0
user@host# set ike gateway hub-to-spoke-gw-1
user@host# set ike ipsec-policy vpn-policy
[edit security ipsec vpn hub-to-spoke-vpn-2]
user@host# set bind-interface st0.1
user@host# set vpn-monitor source-interface ge-0/0/2.0
```

```
user@host# set ike gateway hub-to-spoke-gw-2
user@host# set ike ipsec-policy vpn-policy
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.0
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces ge-0/0/2.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/3.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
```

```
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.1/30;
        }
    }
}
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 10.1.2.1/30;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 10.50.50.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.1/24;
            }
        }
        unit 1 {
            multipoint;
            family inet {
                address 10.20.20.1/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement lan_nw {
    from interface ge-0/0/3.0;
    then accept;
}
[edit]
user@host# show protocols
```

```
bgp {
    group ibgp-1 {
        type internal;
        local-address 10.10.10.1;
        export lan_nw;
        cluster 10.2.3.4;
        allow 10.10.10.0/24;
    }
    group ibgp-2 {
        type internal;
        local-address 10.20.20.1;
        export lan_nw;
        cluster 10.2.3.5;
        allow 10.20.20.0/24;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.2.2.0/30 next-hop 10.1.1.2;
    route 10.3.3.0/30 next-hop 10.1.2.2;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy-1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    policy ike-policy-2 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local2;
        }
    }
    gateway hub-to-spoke-gw-1 {
        ike-policy ike-policy-1;
        dynamic {
            distinguished-name {
                wildcard OU=SLT;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
    gateway hub-to-spoke-gw-2 {
        ike-policy ike-policy-2;
        dynamic {
            distinguished-name {
                wildcard OU=SBU;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/2.0;
    }
[edit]
user@host# show security ipsec
vpn-monitor-options {
    interval 5;
    threshold 2;
}
    proposal ipsec-proposal {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm des-cbc;
    }
    policy vpn-policy {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn hub-to-spoke-vpn-1 {
        bind-interface st0.0;
        vpn-monitor {
            source-interface ge-0/0/1.0;
        }
        ike {
            gateway hub-to-spoke-gw-1;
            ipsec-policy vpn-policy;
        }
    }
    vpn hub-to-spoke-vpn-2 {
        bind-interface st0.1;
        vpn-monitor {
            source-interface ge-0/0/2.0;
        }
        ike {
            gateway hub-to-spoke-gw-2;
            ipsec-policy vpn-policy;
        }
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.0;
        ge-0/0/1.0;
        ge-0/0/2.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/3.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces fe-0/0/1 unit 0 family inet address 10.2.2.1/30
set interfaces fe-0/0/2 unit 0 family inet address 10.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.60.60.1/24
set interfaces st0 unit 0 family inet address 10.10.10.2/24
set interfaces st0 unit 1 family inet address 10.20.20.2/24
set policy-options policy-statement default_route from protocol static
set policy-options policy-statement default_route from route-filter 0.0.0.0/0 exact
set policy-options policy-statement default_route then accept
set policy-options policy-statement lan_nw from interface fe-0/0/4.0
set policy-options policy-statement lan_nw then accept
set protocols bgp group ibgp-1 type internal
set protocols bgp group ibgp-1 local-address 10.10.10.2
set protocols bgp group ibgp-1 export lan_nw
set protocols bgp group ibgp-1 neighbor 10.10.10.1
set protocols bgp group ibgp-2 type internal
set protocols bgp group ibgp-2 local-address 10.20.20.2
set protocols bgp group ibgp-2 export default_route
set protocols bgp group ibgp-2 neighbor 10.20.20.1
set routing-options static route 10.1.1.0/30 next-hop 10.2.2.2
set routing-options static route 10.1.2.0/30 next-hop 10.3.3.2
set routing-options static route 0.0.0.0/0 next-hop st0.1
set routing-options autonomous-system 65010
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy-1 mode main
set security ike policy ike-policy-1 proposals ike-proposal
set security ike policy ike-policy-1 certificate local-certificate Local1
set security ike policy ike-policy-2 mode main
set security ike policy ike-policy-2 proposals ike-proposal
set security ike policy ike-policy-2 certificate local-certificate Local2
set security ike gateway spoke-to-hub-gw-1 ike-policy ike-policy-1
set security ike gateway spoke-to-hub-gw-1 address 10.1.1.1
set security ike gateway spoke-to-hub-gw-1 local-identity distinguished-name
set security ike gateway spoke-to-hub-gw-1 remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw-1 external-interface fe-0/0/1.0
set security ike gateway spoke-to-hub-gw-2 ike-policy ike-policy-2
set security ike gateway spoke-to-hub-gw-2 address 10.1.2.1
set security ike gateway spoke-to-hub-gw-2 local-identity distinguished-name
set security ike gateway spoke-to-hub-gw-2 remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw-2 external-interface fe-0/0/2.0
set security ipsec vpn-monitor-options interval 5
set security ipsec vpn-monitor-options threshold 2
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy proposals ipsec-proposal
set security ipsec vpn spoke-to-hub-1 bind-interface st0.0
set security ipsec vpn spoke-to-hub-1 vpn-monitor destination-ip 10.1.1.1
set security ipsec vpn spoke-to-hub-1 ike gateway spoke-to-hub-gw-1
set security ipsec vpn spoke-to-hub-1 ike ipsec-policy vpn-policy
set security ipsec vpn spoke-to-hub-1 establish-tunnels immediately
set security ipsec vpn spoke-to-hub-2 bind-interface st0.1
set security ipsec vpn spoke-to-hub-2 vpn-monitor destination-ip 10.1.2.1
set security ipsec vpn spoke-to-hub-2 ike gateway spoke-to-hub-gw-2
set security ipsec vpn spoke-to-hub-2 ike ipsec-policy vpn-policy
set security ipsec vpn spoke-to-hub-2 establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces fe-0/0/2.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 1:

1. Configure interfaces.

```
[edit interfaces]
user@host# set fe-0/0/1 unit 0 family inet address 10.2.2.1/30
user@host# set fe-0/0/2 unit 0 family inet address 10.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 10.60.60.1/24
user@host# set st0 unit 0 family inet address 10.10.10.2/24
user@host# set st0 unit 1 family inet address 10.20.20.2/24
```

2. Configure routing protocol.

```
[edit policy-options]
user@host# set policy-statement default_route from protocol static
```

```
user@host# set policy-statement default_route from route-filter 0.0.0.0/0 exact
user@host# set policy-statement default_route then accept
user@host# set policy-statement lan_nw from interface fe-0/0/4.0
user@host# set policy-statement lan_nw then accept
[edit protocols bgp]
user@host# set group ibgp-1 type internal
user@host# set group ibgp-1 local-address 10.10.10.2
user@host# set group ibgp-1 export lan_nw
user@host# set group ibgp-1 neighbor 10.10.10.1
user@host# set group ibgp-2 type internal
user@host# set group ibgp-2 local-address 10.20.20.2
user@host# set group ibgp-2 export default_route
user@host# set group ibgp-2 neighbor 10.20.20.1
[edit routing-options]
user@host# set static route 10.1.1.0/30 next-hop 10.2.2.2
user@host# set static route 10.1.2.0/30 next-hop 10.3.3.2
user@host# set static route 0.0.0.0/0 next-hop st0.1
user@host# set autonomous-system 65010
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy-1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike policy ike-policy-2]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local2
[edit security ike gateway spoke-to-hub-gw-1]
user@host# set ike-policy ike-policy-1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/1.0
[edit security ike gateway spoke-to-hub-gw-2]
user@host# set ike-policy ike-policy-2
```

```
user@host# set address 10.1.2.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/2.0
```

4. Configure Phase 2 options.

```
[edit security ipsec vpn-monitor]
user@host# set options interval 5
user@host# set options threshold 2
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub-1]
user@host# set bind-interface st0.0
user@host# set vpn-monitor destination-ip 10.1.1.1
user@host# set ike gateway spoke-to-hub-gw-1
user@host# set ike ipsec-policy vpn-policy
user@host# set establish-tunnels immediately
[edit security ipsec vpn spoke-to-hub-2]
user@host# set bind-interface st0.1
user@host# set vpn-monitor destination-ip 10.1.2.1
user@host# set ike gateway spoke-to-hub-gw-2
user@host# set ike ipsec-policy vpn-policy
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/1.0
user@host# set interfaces st0.0
user@host# set interfaces fe-0/0/2.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```

```
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
fe-0/0/1 {
    unit 0 {
        family inet {
            address 10.2.2.1/30;
        }
    }
}
    fe-0/0/2 {
        unit 0 {
            family inet {
                address 10.3.3.1/30;
            }
        }
    }
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.60.60.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            family inet {
                address 10.10.10.2/24;
            }
        }
        unit 1 {
            family inet {
                address 10.20.20.2/24;
            }
        }
    }
[edit]
user@host# show policy-options
policy-statement default_route {
    from {
        protocol static;
        route-filter 0.0.0.0/0 exact;
    }
    then accept;
}
    policy-statement lan_nw {
        from interface fe-0/0/4.0;
        then accept;
    }
[edit]
user@host# show protocols
bgp {
    group ibgp-1 {
        type internal;
        local-address 10.10.10.2;
        export lan_nw;
        neighbor 10.10.10.1;
    }
    group ibgp-2 {
        type internal;
        local-address 10.20.20.2;
        export default_route;
        neighbor 10.20.20.1;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.0/30 next-hop 10.2.2.2;
    route 10.1.2.0/30 next-hop 10.3.3.2;
    route 0.0.0.0/0 next-hop st0.1;
    }
autonomous-system 65010;
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy-1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    policy ike-policy-2 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local2;
        }
    }
    gateway spoke-to-hub-gw-1 {
        ike-policy ike-policy-1;
        address 10.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/1.0;
    }
    gateway spoke-to-hub-gw-2 {
        ike-policy ike-policy-2;
        address 10.1.2.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/2.0;
    }
[edit]
user@host# show security ipsec
vpn-monitor-options {
    interval 5;
    threshold 2;
}
    proposal ipsec-proposal {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm des-cbc;
    }
    policy vpn-policy {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub-1 {
        bind-interface st0.0;
        vpn-monitor {
            destination-ip 10.1.1.1;
        }
        ike {
            gateway spoke-to-hub-gw-1;
            ipsec-policy vpn-policy;
        }
        establish-tunnels immediately;
    }
    vpn spoke-to-hub-2 {
        bind-interface st0.1;
        vpn-monitor {
            destination-ip 10.1.2.1;
        }
        ike {
            gateway spoke-to-hub-gw-2;
            ipsec-policy vpn-policy;
        }
        establish-tunnels immediately;
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-0/0/1.0;
        st0.0;
        fe-0/0/2.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying IKE Phase 1 Status (Both Tunnels Are Up)

Purpose

Verify the IKE Phase 1 status when both IPSec VPN tunnels are up.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
3733075 UP     d4f51c28c0a82101  05b125993a864d3c  Main           10.3.3.1
3733076 UP     d53c8a0b7d4c319b  c23c5f7a26388247  Main           10.2.2.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spoke.

Verifying IPsec Phase 2 Status (Both Tunnels Are Up)

Purpose

Verify the IPsec Phase 2 status when both IPsec VPN tunnels are up.

Action

From operational mode, enter the **security ipsec security-associations** command.

```
user@host> security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <268173316 ESP:des/ md5 3cd96946 3555/ unlim U   root 500   10.2.2.1
  >268173316 ESP:des/ md5 1c09b9b 3555/ unlim  U   root 500   10.2.2.1
  <268173313 ESP:des/ md5 7c6ffca3 3340/ unlim U   root 500   10.3.3.1
  >268173313 ESP:des/ md5 33bf6f2f 3340/ unlim U   root 500   10.3.3.1
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spoke.

Verifying IPsec Next-Hop Tunnels (Both Tunnels Are Up)

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway  interface   IPSec VPN name                    Flag     IKE-
ID                            XAUTH username
10.10.10.2        st0.0       hub-to-spoke-vpn-1                Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
10.20.20.2        st0.1       hub-to-spoke-vpn-2                Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SBU, CN=spoke1_backup
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spoke. The next hop should be associated with the correct IPsec VPN name.

Verifying BGP (Both Tunnels Are Up)

Purpose

Verify that BGP references the IP addresses for the `st0` interfaces of the spoke when both IPsec VPN tunnels are up.

Action

From operational mode, enter the **show bgp summary** command.

```
user@host> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Unconfigured peers: 2
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
```

```
Received/Accepted/Damped...
10.10.10.2               65010       5          6       0       0          54
1/1/1/0              0/0/0/0
10.20.20.2               65010      13         16       0       0        4:29
1/1/1/0              0/0/0/0
```

Verifying Learned Routes (Both Tunnels Are Up)

Purpose

Verify that routes to the spoke have been learned when both tunnels are up. The route to 10.60.60.0/24 is through the st0.0 interface and the default route is through the st0.1 interface.

Action

From operational mode, enter the **show route 10.60.60.0** command.

```
user@host> show route 10.60.60.0
inet.0: 48 destinations, 48 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

60.60.60.0/24      *[BGP/170] 00:01:11, localpref 100
                      AS path: I
                    > to 10.10.10.2 via st0.0
```

From operational mode, enter the **show route 0.0.0.0** command.

```
user@host> show route 0.0.0.0
inet.0: 48 destinations, 48 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[BGP/170] 00:04:55, localpref 100
                      AS path: I
                    > to 10.20.20.2 via st0.1
```

Verifying IKE Phase 1 Status (Primary Tunnel Is Down)

Purpose

Verify the IKE Phase 1 status when the primary tunnel is down.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
3733075 UP     d4f51c28c0a82101  05b125993a864d3c  Main           10.3.3.1
3733076 UP     d53c8a0b7d4c319b  c23c5f7a26388247  Main           10.2.2.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spoke.

Verifying IPsec Phase 2 Status (Primary Tunnel Is Down)

Purpose

Verify the IPsec Phase 2 status when the primary tunnel is down.

Action

From operational mode, enter the **security ipsec security-associations** command.

```
user@host> security ipsec security-associations
   Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <268173313 ESP:des/ md5 7c6ffca3 3156/ unlim U   root 500   10.3.3.1
  >268173313 ESP:des/ md5 33bf6f2f 3156/ unlim U   root 500   10.3.3.1
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spoke.

Verifying IPsec Next-Hop Tunnels (Primary Tunnel Is Down)

Purpose

Verify the IPsec next-hop tunnel.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway  interface   IPSec VPN name                    Flag     IKE-
ID                            XAUTH username
10.20.20.2        st0.1       hub-to-spoke-vpn-2                Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SBU, CN=spoke1_backup
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spoke. The next hop should be associated with the correct IPsec VPN name, in this case the backup VPN tunnel.

Verifying BGP (Primary Tunnel Is Down)

Purpose

Verify that BGP references the IP addresses for the `st0` interfaces of the spoke when the primary tunnel is down.

Action

From operational mode, enter the **show bgp summary** command.

```
user@host> show bgp summary
Groups: 2 Peers: 1 Down peers: 0
```

```
Unconfigured peers: 1
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 1          1          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.20.20.2               10         20         24       0       0        7:24
1/1/1/0              0/0/0/0
```

Verifying Learned Routes (Primary Tunnel Is Down)

Purpose

Verify that routes to the spoke have been learned when the primary tunnel is down. Both the route to 10.60.60.0/24 and the default route are through the st0.1 interface.

Action

From operational mode, enter the **show route 10.60.60.0** command.

```
user@host> show route 60.60.60.0
inet.0: 46 destinations, 46 routes (45 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[BGP/170] 00:07:41, localpref 100
                      AS path: I
                    > to 10.20.20.2 via st0.1
```

From operational mode, enter the **show route 0.0.0.0** command.

```
user@host> show route 0.0.0.0
inet.0: 46 destinations, 46 routes (45 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[BGP/170] 00:07:47, localpref 100
                      AS path: I
                    > to 10.20.20.2 via st0.1
```

SEE ALSO

Route-Based IPsec VPNs | 394

Example: Configuring Basic AutoVPN with OSPF

IN THIS SECTION

This example shows how to configure an AutoVPN hub to act as a single termination point, and then configure two spokes to act as tunnels to remote sites. This example configures OSPF to forward packets through the VPN tunnels.

Requirements

This example uses the following hardware and software components:

- Three supported SRX Series Firewalls as AutoVPN hub and spokes
- Junos OS Release 12.1X44-D10 and later that support AutoVPN

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

This example shows the configuration of an AutoVPN hub and the subsequent configurations of two spokes.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). The certificates for the spokes contain the organizational unit (OU) value “SLT” in the subject field; the hub is configured with a group IKE ID to match the value “SLT” in the OU field.

The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and all spokes must have the same values. Table 101 on page 1175 shows the options used in this example.

Table 101: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Basic OSPF Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	2
Authentication algorithm	SHA-1
Encryption algorithm	AES 128 CBC
IKE policy:	
Mode	Main

Table 101: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Basic OSPF Configurations *(Continued)*

Option	Value
IPsec proposal:	
Protocol	ESP
Authentication algorithm	HMAC MD5 96
Encryption algorithm	DES CBC
IPsec policy:	
Perfect Forward Secrecy (PFS) group	14

The same certificate authority (CA) is configured on all devices.

Junos OS only supports a single level of certificate hierarchy.

Table 102 on page 1176 shows the options configured on the hub and on all spokes.

Table 102: AutoVPN Basic OSPF Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	10.1.1.1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string `SLT` in the organizational unit (OU) field	DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate

Table 102: AutoVPN Basic OSPF Configuration for Hub and All Spokes *(Continued)*

Option	Hub	All Spokes
External interface	ge-0/0/1.0	Spoke 1: fe-0/0/1.0 Spoke 2: ge-0/0/1.0
VPN:		
Bind interface	st0.0	st0.0
Establish tunnels	(not configured)	Immediately on configuration commit

Table 103 on page 1177 shows the configuration options that are different on each spoke.

Table 103: Comparison Between the Basic OSPF Spoke Configurations

Option	Spoke 1	Spoke 2
st0.0 interface	10.10.10.2/24	10.10.10.3/24
Interface to internal network	fe-0.0/4.0: 100.60.60.1/24	fe-0.0/4.0: 10.70.70.1/24
Interface to Internet	fe-0/0/1.0: 10.2.2.1/30	ge-0/0/1.0: 10.3.3.1/30

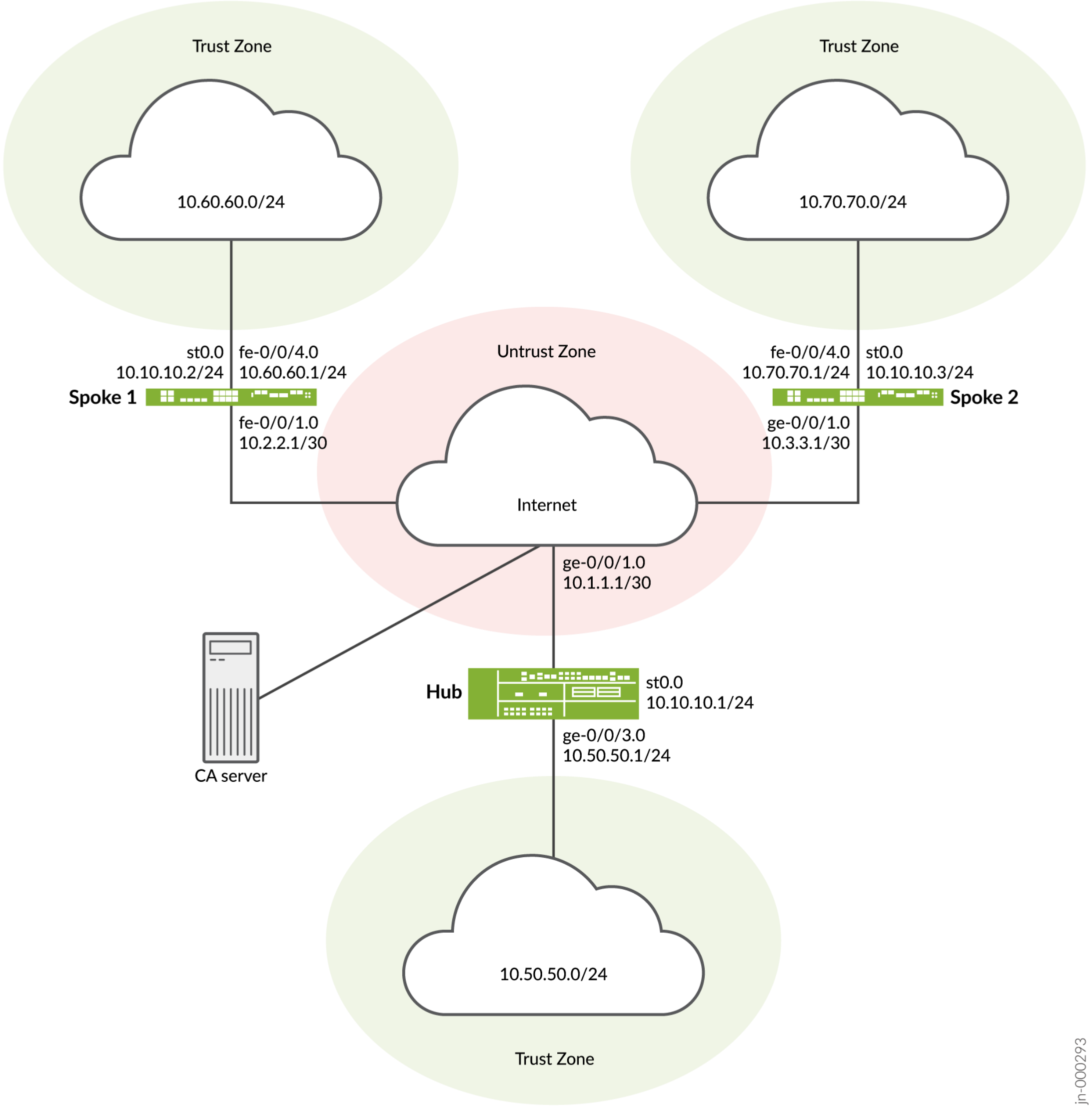
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Topology

Figure 70 on page 1178 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

Figure 70: Basic AutoVPN Deployment with OSPF



Configuration

IN THIS SECTION

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2012 09:39
    Not after: 11- 6-2013 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

   ```
   [edit]
   user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
   user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
   mscep.dll
   user@host# set security pki ca-profile ca-profile1 revocation-check disable
   user@host# commit
   ```

2. Enroll the CA certificate.

   ```
   user@host> request security pki ca-certificate enroll ca-profile ca-profile1
   ```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

   ```
   user@host> request security pki generate-key-pair certificate-id Local1
   ```

4. Enroll the local certificate.

   ```
   user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
   id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
   DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
   ```

5. Verify the local certificate.

   ```
   user@host> show security pki local-certificate detail

   Certificate identifier: Local1
     Certificate version: 3
   ```

```
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2012 09:40
    Not after: 11- 6-2013 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is SLT. The IKE configuration on the hub includes ou=SLT to identify the spoke.

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 2:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
```

```
user@host# set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/
mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

   ```
   user@host> request security pki ca-certificate enroll ca-profile ca-profile1
   ```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

   ```
   user@host> request security pki generate-key-pair certificate-id Local1
   ```

4. Enroll the local certificate.

   ```
   user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
   id Local1 domain-name example.net email spoke2@example.net ip-address 10.3.3.1 subject
   DC=example.net,CN=spoke2,OU=SLT,O=example,L=Tumkur,ST=KA,C=IN challenge-password <password>
   ```

5. Verify the local certificate.

   ```
   user@host> show security pki local-certificate detail

   Certificate identifier: Local1
     Certificate version: 3
     Serial number: 40bb71d400000000258f
     Issuer:
       Common name: CASERVER1, Domain component: net, Domain component: internal
     Subject:
       Organization: example, Organizational unit: SLT, Country: IN, State: KA,
       Locality: Tumkur, Common name: spoke2, Domain component: example.net
     Subject string:
       C=IN, DC=example.net, ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
     Alternate subject: "spoke2@example.net", example.net, 10.3.3.1
     Validity:
       Not before: 11- 6-2012 10:02
       Not after: 11- 6-2013 10:12
   ```

```
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:b6:2e:e2:da:e6:ac:57:e4:5d:ff:de:f6:89
    27:d6:3e:1b:4a:3f:b2:2d:b3:d3:61:ed:ed:6a:07:d9:8a:d2:24:03
    77:1a:fe:84:e1:12:8a:2d:63:6e:bf:02:6b:15:96:5a:4f:37:a0:46
    44:09:96:c0:fd:bb:ab:79:2c:5d:92:bd:31:f0:3b:29:51:ce:89:8e
    7c:2b:02:d0:14:5b:0a:a9:02:93:21:ea:f9:fc:4a:e7:08:bc:b1:6d
    7c:f8:3e:53:58:8e:f1:86:13:fe:78:b5:df:0b:8e:53:00:4a:46:11
    58:4a:38:e9:82:43:d8:25:47:7d:ef:18:f0:ef:a7:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    1a:6d:77:ac:fd:94:68:ce:cf:8a:85:f0:39:fc:e0:6b:fd:fe:b8:66 (sha1)
    00:b1:32:5f:7b:24:9c:e5:02:e6:72:75:9e:a5:f4:77 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/30
set interfaces ge-0/0/3 unit 0 family inet address 10.50.50.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.1/24
set protocols ospf area 0.0.0.0 interface st0.0 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.0 dynamic-neighbors
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set routing-options static route 10.2.2.0/30 next-hop 10.1.1.2
set routing-options static route 10.3.3.0/30 next-hop 10.1.1.2
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway hub-to-spoke-gw ike-policy ike-policy1
set security ike gateway hub-to-spoke-gw dynamic distinguished-name wildcard OU=SLT
set security ike gateway hub-to-spoke-gw dynamic ike-user-type group-ike-id
set security ike gateway hub-to-spoke-gw local-identity distinguished-name
set security ike gateway hub-to-spoke-gw external-interface ge-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn hub-to-spoke-vpn bind-interface st0.0
set security ipsec vpn hub-to-spoke-vpn ike gateway hub-to-spoke-gw
set security ipsec vpn hub-to-spoke-vpn ike ipsec-policy vpn-policy1
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces st0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the hub:

1. Configure the interfaces.

   ```
   [edit interfaces]
   user@host# set ge-0/0/1 unit 0 family inet address 10.1.1.1/30
   user@host# set ge-0/0/3 unit 0 family inet address 10.50.50.1/24
   ```

```
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.1/24
```

2. Configure the routing protocol.

```
[edit protocols ospf]
user@host# set area 0.0.0.0 interface st0.0 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.0 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/3.0
[edit routing-options]
user@host# set static route 2.2.2.0/30 next-hop 10.1.1.2
user@host# set static route 3.3.3.0/30 next-hop 10.1.1.2
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway hub-to-spoke-gw]
user@host# set ike-policy ike-policy1
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
```

```
[edit security ipsec vpn hub-to-spoke-vpn]
user@host# set bind-interface st0.0
user@host# set ike gateway hub-to-spoke-gw
user@host# set ike ipsec-policy vpn-policy1
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/3.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
```

```
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.1/30;
        }
    }
}
    ge-0/0/3 {
        unit 0 {
            family inet {
                address 10.50.50.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.1/24;
            }
        }
    }
[edit]
user@host# show protocols
ospf {
    area 0.0.0.0 {
        interface st0.0 {
            interface-type p2mp;
            dynamic-neighbors;
        }
        interface ge-0/0/3.0;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.2.2.0/30 next-hop 10.1.1.2;
    route 10.3.3.0/30 next-hop 10.1.1.2;
}
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway hub-to-spoke-gw {
        ike-policy ike-policy1;
        dynamic {
            distinguished-name {
                wildcard OU=SLT;
            }
            ike-user-type group-ike-id;
        }
        local-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
[edit]
user@host# show security ipsec
traceoptions {
    flag all;
}
    proposal ipsec-proposal {
        protocol esp;
        authentication-algorithm hmac-md5-96;
        encryption-algorithm des-cbc;
    }
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn hub-to-spoke-vpn {
        bind-interface st0.0;
        ike {
            gateway hub-to-spoke-gw;
            ipsec-policy vpn-policy1;
        }
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.0;
        ge-0/0/1.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/3.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces fe-0/0/1 unit 0 family inet address 10.2.2.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.60.60.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.2/24
set protocols ospf area 0.0.0.0 interface st0.0 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.0 neighbor 10.10.10.1
set protocols ospf area 0.0.0.0 interface fe-0/0/4.0
set routing-options static route 10.1.1.0/30 next-hop 10.2.2.2
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway spoke-to-hub-gw ike-policy ike-policy1
set security ike gateway spoke-to-hub-gw address 10.1.1.1
set security ike gateway spoke-to-hub-gw local-identity distinguished-name
set security ike gateway spoke-to-hub-gw remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw external-interface fe-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn spoke-to-hub bind-interface st0.0
set security ipsec vpn spoke-to-hub ike gateway spoke-to-hub-gw
set security ipsec vpn spoke-to-hub ike ipsec-policy vpn-policy1
set security ipsec vpn spoke-to-hub establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 1:

1. Configure interfaces.

   ```
   [edit interfaces]
   user@host# set fe-0/0/1 unit 0 family inet address 10.2.2.1/30
   user@host# set fe-0/0/4 unit 0 family inet address 10.60.60.1/24
   user@host# set st0 unit 0 multipoint
   user@host# set st0 unit 0 family inet address 10.10.10.2/24
   ```

2. Configure the routing protocol.

   ```
   [edit protocols ospf]
   user@host# set area 0.0.0.0 interface st0.0 interface-type p2mp
   user@host# set area 0.0.0.0 interface st0.0 neighbor 10.10.10.1
   user@host# set area 0.0.0.0 interface fe-0/0/4.0
   [edit routing-options]
   user@host# set static route 10.1.1.0/30 next-hop 10.2.2.2
   ```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway spoke-to-hub-gw]
user@host# set ike-policy ike-policy1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface fe-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub]
user@host# set bind-interface st0.0
user@host# set ike gateway spoke-to-hub-gw
user@host# set ike ipsec-policy vpn-policy1
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/1.0
user@host# set interfaces st0.0
```

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
fe-0/0/1 {
    unit 0 {
        family inet {
            address 10.2.2.1/30;
        }
    }
}
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.60.60.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.2/24;
            }
        }
    }
[edit]
user@host# show protocols
ospf {
    area 0.0.0.0 {
        interface st0.0 {
            interface-type p2mp;
            neighbor 10.10.10.1;
        }
        interface fe-0/0/4.0;
    }
}
[edit]
user@host# show routing-options
static {
    route 10.1.1.0/30 next-hop 10.2.2.2;
}
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway spoke-to-hub-gw {
        ike-policy ike-policy1;
        address 10.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface fe-0/0/1.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub {
        bind-interface st0.0;
        ike {
            gateway spoke-to-hub-gw;
            ipsec-policy vpn-policy1;
        }
        establish-tunnels immediately;
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-0/0/1.0;
        st0.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.3.3.1/30
set interfaces fe-0/0/4 unit 0 family inet address 10.70.70.1/24
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet address 10.10.10.3/24
set protocols ospf area 0.0.0.0 interface st0.0 interface-type p2mp
set protocols ospf area 0.0.0.0 interface st0.0 neighbor 10.10.10.1
set protocols ospf area 0.0.0.0 interface fe-0/0/4.0
set routing-options static route 10.1.1.1/32 next-hop 10.3.3.2
set security ike proposal ike-proposal authentication-method rsa-signatures
set security ike proposal ike-proposal dh-group group2
set security ike proposal ike-proposal authentication-algorithm sha1
set security ike proposal ike-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-policy1 mode main
set security ike policy ike-policy1 proposals ike-proposal
set security ike policy ike-policy1 certificate local-certificate Local1
set security ike gateway spoke-to-hub-gw ike-policy ike-policy1
set security ike gateway spoke-to-hub-gw address 10.1.1.1
set security ike gateway spoke-to-hub-gw local-identity distinguished-name
set security ike gateway spoke-to-hub-gw remote-identity distinguished-name
set security ike gateway spoke-to-hub-gw external-interface ge-0/0/1.0
set security ipsec proposal ipsec-proposal protocol esp
set security ipsec proposal ipsec-proposal authentication-algorithm hmac-md5-96
set security ipsec proposal ipsec-proposal encryption-algorithm des-cbc
set security ipsec policy vpn-policy1 perfect-forward-secrecy keys group14
set security ipsec policy vpn-policy1 proposals ipsec-proposal
set security ipsec vpn spoke-to-hub bind-interface st0.0
set security ipsec vpn spoke-to-hub ike gateway spoke-to-hub-gw
set security ipsec vpn spoke-to-hub ike ipsec-policy vpn-policy1
set security ipsec vpn spoke-to-hub establish-tunnels immediately
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces st0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-0/0/4.0
set security policies default-policy permit-all
set security pki ca-profile ca-profile1 ca-identity ca-profile1
set security pki ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
set security pki ca-profile ca-profile1 revocation-check disable
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 2:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.3.3.1/30
user@host# set fe-0/0/4 unit 0 family inet address 10.70.70.1/24
user@host# set st0 unit 0 multipoint
user@host# set st0 unit 0 family inet address 10.10.10.3/24
```

2. Configure the routing protocol.

```
[edit protocols ospf]
user@host# set area 0.0.0.0 interface st0.0 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.0 neighbor 10.10.10.1
user@host# set area 0.0.0.0 interface fe-0/0/4.0
[edit routing-options]
user@host# set static route 10.1.1.1/32 next-hop 10.3.3.2
```

3. Configure Phase 1 options.

```
[edit security ike proposal ike-proposal]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-128-cbc
[edit security ike policy ike-policy1]
user@host# set mode main
user@host# set proposals ike-proposal
user@host# set certificate local-certificate Local1
[edit security ike gateway spoke-to-hub-gw]
user@host# set ike-policy ike-policy1
user@host# set address 10.1.1.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name
user@host# set external-interface ge-0/0/1.0
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal ipsec-proposal]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-md5-96
user@host# set encryption-algorithm des-cbc
[edit security ipsec policy vpn-policy1]
user@host# set perfect-forward-secrecy keys group14
user@host# set proposals ipsec-proposal
[edit security ipsec vpn spoke-to-hub]
user@host# set bind-interface st0.0
user@host# set ike gateway spoke-to-hub-gw
user@host# set ike ipsec-policy vpn-policy1
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
user@host# set interfaces st0.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces fe-0/0/4.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ca-profile1 ca-identity ca-profile1
user@host# set ca-profile ca-profile1 enrollment url http://pc4/certsrv/mscep/mscep.dll
user@host# set ca-profile ca-profile1 revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.3.3.1/30;
        }
    }
}
    fe-0/0/4 {
        unit 0 {
            family inet {
                address 10.70.70.1/24;
            }
        }
    }
    st0 {
        unit 0 {
            multipoint;
            family inet {
                address 10.10.10.3/24;
            }
        }
    }
[edit]
user@host# show protocols
ospf {
    area 0.0.0.0 {
        interface st0.0 {
            interface-type p2mp;
            neighbor 10.10.10.1;
        }
        interface fe-0/0/4.0;
    }
}
```

```
[edit]
user@host# show routing-options
static {
    route 10.1.1.1/32 next-hop 10.3.3.2;
}
[edit]
user@host# show security ike
proposal ike-proposal {
    authentication-method rsa-signatures;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
}
    policy ike-policy1 {
        mode main;
        proposals ike-proposal;
        certificate {
            local-certificate Local1;
        }
    }
    gateway spoke-to-hub-gw {
        ike-policy ike-policy1;
        address 10.1.1.1;
        local-identity distinguished-name;
        remote-identity distinguished-name;
        external-interface ge-0/0/1.0;
    }
[edit]
user@host# show security ipsec
proposal ipsec-proposal {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm des-cbc;
}
    policy vpn-policy1 {
        perfect-forward-secrecy {
            keys group14;
        }
        proposals ipsec-proposal;
    }
    vpn spoke-to-hub {
        bind-interface st0.0;
        ike {
            gateway spoke-to-hub-gw;
            ipsec-policy vpn-policy1;
        }
        establish-tunnels immediately;
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            fe-0/0/4.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ca-profile1 {
    ca-identity ca-profile1;
    enrollment {
        url http://pc4/certsrv/mscep/mscep.dll;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying IKE Phase 1 Status

Purpose

Verify the IKE Phase 1 status.

Action

From operational mode, enter the **show security ike security-associations** command.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode          Remote Address
5480159 UP     22432fb6f7fbc389  412b751f79b45099  Main          10.2.2.1
5480161 UP     d455050707bc3eaf  b3dde111232270d2  Main          10.3.3.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

Verifying IPsec Phase 2 Status

Purpose

Verify the IPsec Phase 2 status.

Action

From operational mode, enter the **security ipsec security-associations** command.

```
user@host> security ipsec security-associations
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <268173400 ESP:des/ md5 f38eea12 2954/ unlim -   root 500   10.2.2.1
  >268173400 ESP:des/ md5 bb48d228 2954/ unlim -   root 500   10.2.2.1
  <268173401 ESP:des/ md5 bcd1390b 3530/ unlim -   root 500   10.3.3.1
  >268173401 ESP:des/ md5 77fcf6e2 3530/ unlim -   root 500   10.3.3.1
```

Meaning

The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway  interface   IPSec VPN name                    Flag     IKE-
ID                            XAUTH username
10.10.10.2        st0.0       hub-to-spoke-vpn                  Auto     C=IN, DC=example.net,
ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
10.10.10.3        st0.0       hub-to-spoke-vpn                  Auto     C=IN, DC=example.net,
ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying OSPF

Purpose

Verify that OSPF references the IP addresses for the `st0` interfaces of the spokes.

Action

From operational mode, enter the **show ospf neighbor** command.

```
user@host> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
10.10.10.3       st0.0                  Full      10.255.226.179   128    32
10.10.10.2       st0.0                  Full      10.207.36.182    128    38
```

Verifying Learned Routes

Purpose

Verify that routes to the spokes have been learned.

Action

From operational mode, enter the **show route 60.60.60.0** command.

```
user@host> show route 10.60.60.0
 inet.0: 48 destinations, 48 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.60.60.0/24      *[OSPF/10] 00:51:13, metric 2
                    > to 10.10.10.2 via st0.0
```

From operational mode, enter the **show route 10.70.70.0** command.

```
user@host> show route 10.70.70.0
inet.0: 48 destinations, 48 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.70.70.0/24      *[OSPF/10] 00:51:48, metric 2
                    > to 10.10.10.3 via st0.0
```

SEE ALSO

Route-Based IPsec VPNs | 394

Example: Configuring AutoVPN with OSPFv3 for IPv6 Traffic

IN THIS SECTION

This example shows how to configure an AutoVPN hub to act as a single termination point, and then configure two spokes to act as tunnels to remote sites. This example configures AutoVPN for IPv6 environment using OSPFv3 to forward packets through the VPN tunnels.

Requirements

This example uses the following hardware and software components:

- Three supported SRX Series Firewalls as AutoVPN hub and spokes.
- Junos OS Release 18.1R1 and later releases.

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

You should be familiar with the dynamic routing protocol that is used to forward packets through the VPN tunnels.

Overview

IN THIS SECTION

This example shows the configuration of an AutoVPN with OSPFv3 routing protocol on hub and the subsequent configurations of two spokes.

In this example, the first step is to enroll digital certificates in each device using the Simple Certificate Enrollment Protocol (SCEP). The certificates for the spokes contain the organizational unit (OU) value “SLT” in the subject field; the hub is configured with a group IKE ID to match the value “SLT” in the OU field.

The spokes establish IPsec VPN connections to the hub, which allows them to communicate with each other as well as access resources on the hub. Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hub and all spokes must have the same values. Table 104 on page 1209 shows the options used in this example.

Table 104: Phase 1 and Phase 2 Options for AutoVPN Hub and Spoke Basic OSPFv3 Configurations

Option	Value
IKE proposal:	
Authentication method	RSA digital certificates
Diffie-Hellman (DH) group	19
Authentication algorithm	SHA-384
Encryption algorithm	AES 256 CBC
IKE policy:	
Mode	Main
IPsec proposal:	
Protocol	ESP
Lifetime seconds	3000
Encryption algorithm	AES 256 GCM
IPsec policy:	
Perfect Forward Secrecy (PFS) group	19

The same certificate authority (CA) is configured on all devices.

Table 105 on page 1210 shows the options configured on the hub and on all spokes.

Table 105: AutoVPN OSPFv3 Configuration for Hub and All Spokes

Option	Hub	All Spokes
IKE gateway:		
Remote IP address	Dynamic	2001:db8:2000::1
Remote IKE ID	Distinguished name (DN) on the spoke's certificate with the string SLT in the organizational unit (OU) field	DN on the hub's certificate
Local IKE ID	DN on the hub's certificate	DN on the spoke's certificate
External interface	ge-0/0/0	Spoke 1: ge-0/0/0.0 Spoke 2: ge-0/0/0.0
VPN:		
Bind interface	st0.1	st0.1
Establish tunnels	(not configured)	Immediately on configuration commit

Table 106 on page 1210 shows the configuration options that are different on each spoke.

Table 106: Comparison Between the OSPFv3 Spoke Configurations

Option	Spoke 1	Spoke 2
st0.1 interface	2001:db8:7000::2/64	2001:db8:7000::3/64
Interface to internal network	(ge-0/0/1.0) 2001:db8:4000::1/64	(ge-0/0/1.0) 2001:db8:6000::1/64
Interface to Internet	(ge-0/0/0.0) 2001:db8:3000::2/64	(ge-0/0/0.0) 2001:db8:5000::2/64

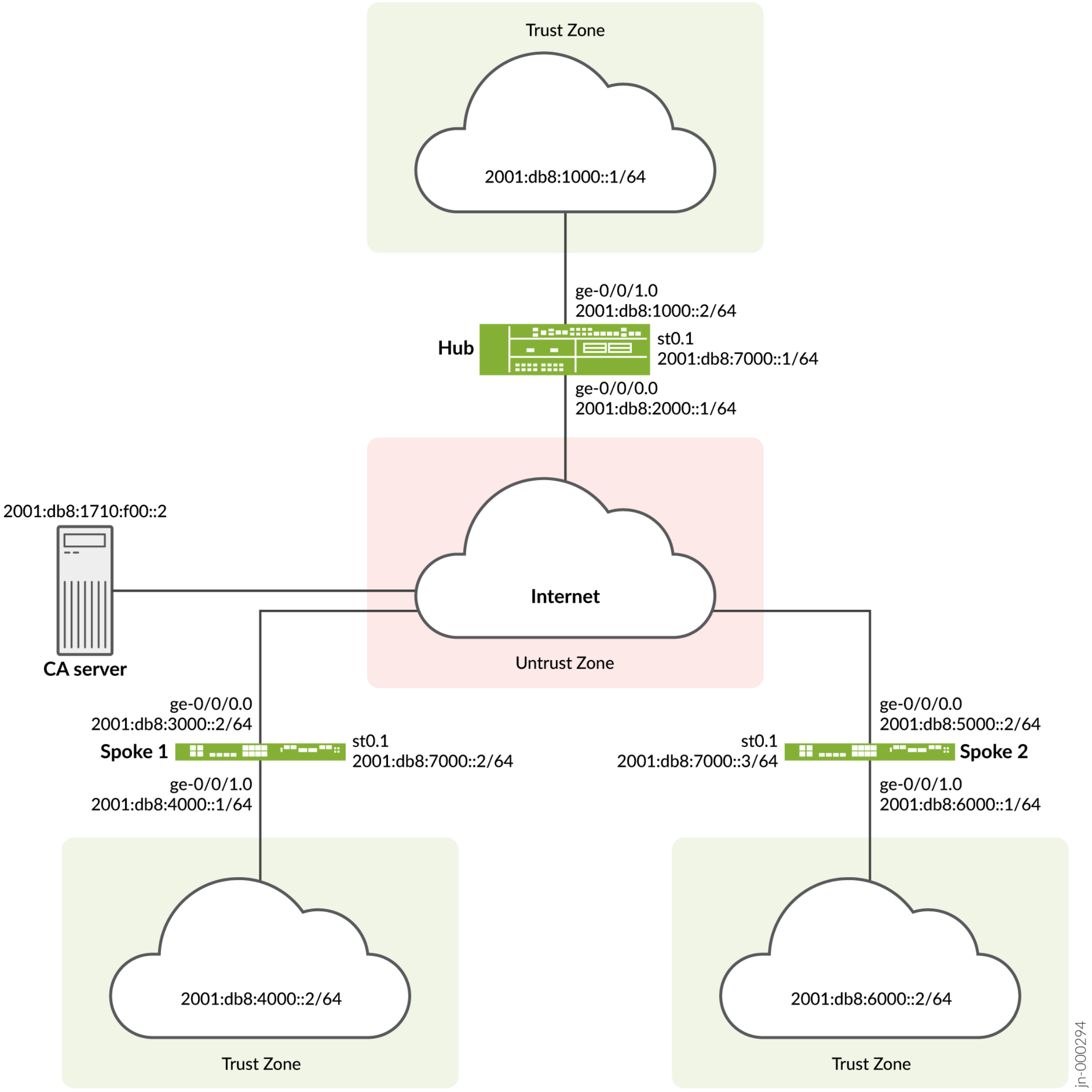
Routing information for all devices is exchanged through the VPN tunnels.

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*.

Topology

Figure 71 on page 1211 shows the SRX Series Firewalls to be configured for AutoVPN in this example.

Figure 71: Basic AutoVPN Deployment with OSPFv3



Configuration

IN THIS SECTION

To configure AutoVPN, perform these tasks:

The first section describes how to obtain CA and local certificates online using the Simple Certificate Enrollment Protocol (SCEP) on the hub and spoke devices.

Enroll Device Certificates with SCEP

Step-by-Step Procedure

To enroll digital certificates with SCEP on the hub:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email hub@example.net ip-address 10.1.1.1 subject
DC=example.net,CN=hub,OU=SLT,O=example,L=Bengaluru,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a6d5f300000000258d
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Bengaluru, Common name: hub, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Bengaluru, O=example, OU=SLT, CN=hub
  Alternate subject: "hub@example.net", example.net, 10.1.1.1
  Validity:
    Not before: 11- 6-2020 09:39
    Not after: 11- 6-2021 09:49
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:c9:c9:cc:30:b6:7a:86:12:89:b5:18:b3:76
    01:2d:cc:65:a8:a8:42:78:cd:d0:9a:a2:c0:aa:c4:bd:da:af:88:f3
    2a:78:1f:0a:58:e6:11:2c:81:8f:0e:7c:de:86:fc:48:4c:28:5b:8b
    34:91:ff:2e:91:e7:b5:bd:79:12:de:39:46:d9:fb:5c:91:41:d1:da
    90:f5:09:00:9b:90:07:9d:50:92:7d:ff:fb:3f:3c:bc:34:e7:e3:c8
    ea:cb:99:18:b4:b6:1d:a8:99:d3:36:b9:1b:36:ef:3e:a1:fd:48:82
    6a:da:22:07:da:e0:d2:55:ef:57:be:09:7a:0e:17:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    e1:f7:a1:a6:1e:c3:97:69:a5:07:9b:09:14:1a:c7:ae:09:f1:f6:35 (sha1)
    a0:02:fa:8d:5c:63:e5:6d:f7:f4:78:56:ac:4e:b2:c4 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 1:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke1@example.net ip-address 10.2.2.1 subject
DC=example.net,CN=spoke1,OU=SLT,O=example,L=Mysore,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40a7975f00000000258e
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
    Organization: example, Organizational unit: SLT, Country: IN, State: KA,
    Locality: Mysore, Common name: spoke1, Domain component: example.net
  Subject string:
    C=IN, DC=example.net, ST=KA, L=Mysore, O=example, OU=SLT, CN=spoke1
  Alternate subject: "spoke1@example.net", example.net, 10.2.2.1
  Validity:
    Not before: 11- 6-2020 09:40
    Not after: 11- 6-2021 09:50
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:d8:45:09:77:cd:36:9a:6f:58:44:18:91:db
    b0:c7:8a:ee:c8:d7:a6:d2:e2:e7:20:46:2b:26:1a:92:e2:4e:8a:ce
    c9:25:d9:74:a2:81:ad:ea:e0:38:a0:2f:2d:ab:a6:58:ac:88:35:f4
    90:01:08:33:33:75:2c:44:26:f8:25:18:97:96:e4:28:de:3b:35:f2
    4a:f5:92:b7:57:ae:73:4f:8e:56:71:ab:81:54:1d:75:88:77:13:64
    1b:6b:01:96:15:0a:1c:54:e3:db:f8:ec:ec:27:5b:86:39:c1:09:a1
    e4:24:1a:19:0d:14:2c:4b:94:a4:04:91:3f:cb:ef:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    http://ca-server1/CertEnroll/CASERVER1.crl
    file://\\ca-server1\CertEnroll\CASERVER1.crl
  Fingerprint:
    b6:24:2a:0e:96:5d:8c:4a:11:f3:5a:24:89:7c:df:ea:d5:c0:80:56 (sha1)
    31:58:7f:15:bb:d4:66:b8:76:1a:42:4a:8a:16:b3:a9 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Step-by-Step Procedure

To enroll digital certificates with SCEP on spoke 2:

1. Configure the CA.

```
[edit]
user@host# set security pki ca-profile ca-profile1 ca-identity ca-profile1
user@host# set security pki ca-profile ca-profile1 enrollment url http://2001:db8:1710:f00::2/
certsrv/mscep/mscep.dll
user@host# set security pki ca-profile ca-profile1 revocation-check disable
user@host# commit
```

2. Enroll the CA certificate.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile1
```

   Type **yes** at the prompt to load the CA certificate.

3. Generate a key pair.

```
user@host> request security pki generate-key-pair certificate-id Local1
```

4. Enroll the local certificate.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile1 certificate-
id Local1 domain-name example.net email spoke2@example.net ip-address 10.3.3.1 subject
DC=example.net,CN=spoke2,OU=SLT,O=example,L=Tumkur,ST=KA,C=IN challenge-password <password>
```

5. Verify the local certificate.

```
user@host> show security pki local-certificate detail

Certificate identifier: Local1
  Certificate version: 3
  Serial number: 40bb71d400000000258f
  Issuer:
    Common name: CASERVER1, Domain component: net, Domain component: internal
  Subject:
```

```
  Organization: example, Organizational unit: SLT, Country: IN, State: KA,
  Locality: Tumkur, Common name: spoke2, Domain component: example.net
Subject string:
  C=IN, DC=example.net, ST=KA, L=Tumkur, O=example, OU=SLT, CN=spoke2
Alternate subject: "spoke2@example.net", example.net, 10.3.3.1
Validity:
  Not before: 11- 6-2020 10:02
  Not after: 11- 6-2021 10:12
Public key algorithm: rsaEncryption(1024 bits)
  30:81:89:02:81:81:00:b6:2e:e2:da:e6:ac:57:e4:5d:ff:de:f6:89
  27:d6:3e:1b:4a:3f:b2:2d:b3:d3:61:ed:ed:6a:07:d9:8a:d2:24:03
  77:1a:fe:84:e1:12:8a:2d:63:6e:bf:02:6b:15:96:5a:4f:37:a0:46
  44:09:96:c0:fd:bb:ab:79:2c:5d:92:bd:31:f0:3b:29:51:ce:89:8e
  7c:2b:02:d0:14:5b:0a:a9:02:93:21:ea:f9:fc:4a:e7:08:bc:b1:6d
  7c:f8:3e:53:58:8e:f1:86:13:fe:78:b5:df:0b:8e:53:00:4a:46:11
  58:4a:38:e9:82:43:d8:25:47:7d:ef:18:f0:ef:a7:02:03:01:00:01
Signature algorithm: sha1WithRSAEncryption
Distribution CRL:
  http://ca-server1/CertEnroll/CASERVER1.crl
  file://\\ca-server1\CertEnroll\CASERVER1.crl
Fingerprint:
  1a:6d:77:ac:fd:94:68:ce:cf:8a:85:f0:39:fc:e0:6b:fd:fe:b8:66 (sha1)
  00:b1:32:5f:7b:24:9c:e5:02:e6:72:75:9e:a5:f4:77 (md5)
Auto-re-enrollment:
  Status: Disabled
  Next trigger time: Timer not started
```

The organizational unit (OU) shown in the subject field is `SLT`. The IKE configuration on the hub includes `ou=SLT` to identify the spoke.

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate HUB
set security ike gateway IKE_GWA_1 ike-policy IKE_POL
set security ike gateway IKE_GWA_1 dynamic distinguished-name wildcard OU=SLT
set security ike gateway IKE_GWA_1 dead-peer-detection always-send
set security ike gateway IKE_GWA_1 dead-peer-detection interval 10
set security ike gateway IKE_GWA_1 dead-peer-detection threshold 3
set security ike gateway IKE_GWA_1 local-identity distinguished-name
set security ike gateway IKE_GWA_1 external-interface ge-0/0/0
set security ike gateway IKE_GWA_1 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPNA_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPNA_1 ike gateway IKE_GWA_1
set security ipsec vpn IPSEC_VPNA_1 ike ipsec-policy IPSEC_POL
set security policies default-policy permit-all
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/l..0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:2000::1/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1000::2/64
set interfaces st0 unit 1 multipoint
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::1/64
set routing-options rib inet6.0 static route 2001:db8:3000::/64 next-hop 2001:db8:2000::1
set routing-options rib inet6.0 static route 2001:db8:5000::/64 next-hop 2001:db8:2000::1
set protocols ospf3 traceoptions file ospf
set protocols ospf3 traceoptions flag all
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf3 area 0.0.0.0 interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the hub:

1. Configure the interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:2000::1/64
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:1000::2/64
user@host# set st0 unit 1 multipoint
user@host# set st0 unit 1 family inet6 address 2001:db8:7000::1/64
```

2. Configure the routing protocol.

```
[edit protocols ospf3]
user@host# set traceoptions file ospf
user@host# set traceoptions flag all
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/1.0
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:3000::/64 next-hop 2001:db8:2000::1
user@host# set rib inet6.0 static route 2001:db8:5000::/64 next-hop 2001:db8:2000::1
```

3. Configure Phase 1 options.

```
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike proposal IKE_PROP]
```

```
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate HUB
[edit security ike gateway IKE_GWA_1]
user@host# set ike-policy IKE_POL
user@host# set dynamic distinguished-name wildcard OU=SLT
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROP]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPNA_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GWA_1
user@host# set ike ipsec-policy IPSEC_POL
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
user@host# set interfaces st0.1
[edit security zones security-zone trust]
```

```
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/1.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set pki ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:2000::1/64;
        }
    }
}
    ge-0/0/1 {
        unit 0 {
            family inet6 {
                address 2001:db8:1000::2/64;
            }
        }
    }
    st0 {
        unit 1 {
            family inet6 {
                address 2001:db8:7000::1/64;
            }
        }
    }
[edit]
user@host# show protocols
ospf3 {
    traceoptions {
        file ospf;
        flag all;
    }
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            demand-circuit;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
    route 2001:db8:3000::/64 next-hop 2001:db8::1;
    route 2001:db8:5000::/64 next-hop 2001:db8::1;
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
    policy IKE_POL {
        mode main;
        proposals IKE_PROP;
        certificate {
            local-certificate HUB;
        }
    }
    gateway IKE_GWA_1 {
        ike-policy IKE_POL;
        dynamic {
            distinguished-name {
                wildcard OU=SLT;
                }
            }
            dead-peer-detection {
                always-send;
                interval 10;
                threshold 3;
            }
        local-identity distinguished-name;
        external-interface ge-0/0/0.0;
        version v1-only;
    }
[edit]
user@host# show security ipsec
    proposal IPSEC_PROP {
        protocol esp;
        authentication-algorithm aes-256-gcm;
        set lifetime-seconds 3000;
    }
    policy IPSEC_POL {
        perfect-forward-secrecy {
            keys group19;
        }
        proposals IPSEC_PROP;
    }
    vpn IPSEC_VPNA_1 {
        bind-interface st0.1;
        ike {
            gateway IKE_GWA_1;
            ipsec-policy IPSEC_POL;
        }
    }
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/0.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                ospf3;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE1
set security ike gateway IKE_GW_SPOKE_1 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_1 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_1 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_1 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_1 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_1 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_1 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_1 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike gateway IKE_GW_SPOKE_1
set security ipsec vpn IPSEC_VPN_SPOKE_1 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_1 establish-tunnels immediately
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::2/64
set routing-options rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:3000::2
set protocols ospf3 traceoptions file ospf
set protocols ospf3 traceoptions flag all
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf3 area 0.0.0.0 interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 1:

1. Configure interfaces.

   ```
   [edit interfaces]
   user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:3000::2/64
   ```

```
user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:4000::1/64
user@host# set st0 unit 1 family inet6 address 2001:db8:7000::2/64
```

2. Configure the routing protocol.

```
[edit protocols ospf3]
user@host# set traceoptions file ospf
user@host# set traceoptions flag all
user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
user@host# set area 0.0.0.0 interface st0.1 demand-circuit
user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
user@host# set area 0.0.0.0 interface ge-0/0/1.0
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:3000::2
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE1
[edit security ike gateway IKE_GW_SPOKE_1]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0.0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROPl]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPN_SPOKE_1]
user@host# set bind-interface st0.1
user@host# set ike gateway IKE_GW_SPOKE_1
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels immediately
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/1.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
```

```
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show security pki` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:3000::2/64;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:db8:4000::1/64;
        }
    }
}
    st0 {
        unit 1 {
            family inet6 {
                address 2001:db8:7000::2/64;
            }
        }
    }
[edit]
user@host# show protocols
ospf3 {
    traceoptions {
        file ospf;
        flag all;
    }
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            demand-circuit;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
    }
}
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
    route 2001:db8:2000::/64 next-hop [ 2001:db8:3000::1 2001:db8:5000::1 ];
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE1;
    }
}
gateway IKE_GW_SPOKE_1 {
    ike-policy IKE_POL;
    address 2001:db8:2000::1;
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=SLT;
    external-interface ge-0/0/0.0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPN_SPOKE_1 {
    bind-interface st0.1;
    ike {
        gateway IKE_GW_SPOKE_1;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.1;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                ospf3;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Spoke 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security pki ca-profile ROOT-CA ca-identity ROOT-CA
set security pki ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
set security pki ca-profile ROOT-CA enrollment retry 5
set security pki ca-profile ROOT-CA enrollment retry-interval 0
set security pki ca-profile ROOT-CA revocation-check disable
set security ike traceoptions file ik
set security ike traceoptions flag all
set security ike proposal IKE_PROP authentication-method rsa-signatures
set security ike proposal IKE_PROP dh-group group19
set security ike proposal IKE_PROP authentication-algorithm sha-384
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 6000
set security ike policy IKE_POL mode main
set security ike policy IKE_POL proposals IKE_PROP
set security ike policy IKE_POL certificate local-certificate SPOKE2
set security ike gateway IKE_GW_SPOKE_2 ike-policy IKE_POL
set security ike gateway IKE_GW_SPOKE_2 address 2001:db8:2000::1
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection always-send
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection interval 10
set security ike gateway IKE_GW_SPOKE_2 dead-peer-detection threshold 3
set security ike gateway IKE_GW_SPOKE_2 local-identity distinguished-name
set security ike gateway IKE_GW_SPOKE_2 remote-identity distinguished-name container OU=SLT
set security ike gateway IKE_GW_SPOKE_2 external-interface ge-0/0/0.0
set security ike gateway IKE_GW_SPOKE_2 version v1-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3000
set security ipsec policy IPSEC_POL perfect-forward-secrecy keys group19
set security ipsec policy IPSEC_POL proposals IPSEC_PROP
set security ipsec vpn IPSEC_VPN_SPOKE_2 bind-interface st0.1
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike gateway IKE_GW_SPOKE_2
set security ipsec vpn IPSEC_VPN_SPOKE_2 ike ipsec-policy IPSEC_POL
set security ipsec vpn IPSEC_VPN_SPOKE_2 establish-tunnels on-traffic
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols ospf3
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols ospf3
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
set interfaces st0 unit 1 family inet6 address 2001:db8:7000::3/64
set routing-options rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
set protocols ospf3 traceoptions file ospf
set protocols ospf3 traceoptions flag all
set protocols ospf3 area 0.0.0.0 interface st0.1 interface-type p2mp
set protocols ospf3 area 0.0.0.0 interface st0.1 demand-circuit
set protocols ospf3 area 0.0.0.0 interface st0.1 dynamic-neighbors
set protocols ospf3 area 0.0.0.0 interface ge-0/0/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure spoke 2:

1. Configure interfaces.

   ```
   [edit interfaces]
   user@host# set ge-0/0/0 unit 0 family inet6 address 2001:db8:5000::2/64
   user@host# set ge-0/0/1 unit 0 family inet6 address 2001:db8:6000::1/64
   user@host# set st0 unit 1 family inet6 address 2001:db8:7000::3/64
   ```

2. Configure the routing protocol.

   ```
   [edit protocols ospf3]
   user@host# set traceoptions file ospf
   user@host# set traceoptions flag all
   user@host# set area 0.0.0.0 interface st0.1 interface-type p2mp
   user@host# set area 0.0.0.0 interface st0.1 demand-circuit
   user@host# set area 0.0.0.0 interface st0.1 dynamic-neighbors
   ```

```
user@host# set area 0.0.0.0 interface ge-0/0/1.0
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:2000::/64 next-hop 2001:db8:5000::1
```

3. Configure Phase 1 options.

```
[edit security ike proposal IKE_PROP]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group19
user@host# set authentication-algorithm sha-384
user@host# set encryption-algorithm aes-256-cbc
user@host# set lifetime-seconds 6000
[edit security ike traceoptions]
user@host# set file ik
user@host# set flag all
[edit security ike policy IKE_POL]
user@host# set mode main
user@host# set proposals IKE_PROP
user@host# set certificate local-certificate SPOKE2
[edit security ike gateway IKE_GW_SPOKE_2]
user@host# set ike-policy IKE_POL
user@host# set address 2001:db8:2000::1
user@host# set dead-peer-detection always-send
user@host# set dead-peer-detection interval 10
user@host# set dead-peer-detection threshold 3
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container OU=SLT
user@host# set external-interface ge-0/0/0.0
user@host# set version v1-only
```

4. Configure Phase 2 options.

```
[edit security ipsec proposal IPSEC_PROPl]
user@host# set protocol esp
user@host# set encryption-algorithm aes-256-gcm
user@host# set lifetime-seconds 3000
[edit security ipsec policy IPSEC_POL]
user@host# set perfect-forward-secrecy keys group19
user@host# set proposals IPSEC_PROP
[edit security ipsec vpn IPSEC_VPN_SPOKE_2]
user@host# set bind-interface st0.1
```

```
user@host# set ike gateway IKE_GW_SPOKE_2
user@host# set ike ipsec-policy IPSEC_POL
user@host# set establish-tunnels on-traffic
```

5. Configure zones.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/1.0
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols ospf3
user@host# set interfaces ge-0/0/0.0
```

6. Configure the default security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure the CA profile.

```
[edit security pki]
user@host# set ca-profile ROOT-CA ca-identity ROOT-CA
user@host# set ca-profile ROOT-CA enrollment url http://2001:db8:1710:f00::2/certsrv/mscep/
mscep.dll
user@host# set ca-profile ROOT-CA enrollment retry 5
user@host# set ca-profile ROOT-CA enrollment retry-interval 0
user@host# set ca-profile ROOT-CA revocation-check disable
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, `show routing-options`, `show security ike`, `show security ipsec`, `show security zones`, `show security policies`, and `show` security pki commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:db8:5000::2/64;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:db8:6000::1/64;
        }
    }
}
    st0 {
        unit 1 {
            family inet6 {
                address 2001:db8:7000::3/64;
            }
        }
    }
[edit]
user@host# show protocols
ospf3 {
    traceoptions {
        file ospf;
        flag all;
    }
    area 0.0.0.0 {
        interface st0.1 {
            interface-type p2mp;
            demand-circuit;
            dynamic-neighbors;
        }
        interface ge-0/0/1.0;
    }
}
```

```
[edit]
user@host# show routing-options
rib inet6.0 {
    static {
    route 2001:db8:2000::/64 next-hop [ 2001:db8:3000::1 2001:db8:5000::1 ];
    }
}
[edit]
user@host# show security ike
traceoptions {
    file ik;
    flag all;
}
proposal IKE_PROP {
    authentication-method rsa-signatures;
    dh-group group19;
    authentication-algorithm sha-384;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 6000;
}
policy IKE_POL {
    mode main;
    proposals IKE_PROP;
    certificate {
        local-certificate SPOKE2;
    }
}
gateway IKE_GW_SPOKE_2 {
    ike-policy IKE_POL;
    address 2001:db8:2000::1;
    dead-peer-detection {
        always-send;
        interval 10;
        threshold 3;
    }
    local-identity distinguished-name;
    remote-identity distinguished-name container OU=SLT;
    external-interface ge-0/0/0.0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3000;
}
policy IPSEC_POL {
    perfect-forward-secrecy {
        keys group19;
    }
    proposals IPSEC_PROP;
}
vpn IPSEC_VPN_SPOKE_2 {
    bind-interface st0.1;
    ike {
        gateway IKE_GW_SPOKE_2;
        ipsec-policy IPSEC_POL;
    }
    establish-tunnels on-traffic;
}
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            ospf3;
        }
    }
    interfaces {
        ge-0/0/1.0;
        st0.0;
    }
}
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                ospf3;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
[edit]
user@host# show security pki
ca-profile ROOT-CA {
    ca-identity ROOT-CA;
    enrollment {
        url http://2001:db8:1710:f00::2/certsrv/mscep/mscep.dll;
        retry 5;
        retry-interval 0;
    }
    revocation-check {
        disable;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying IKE Status

Purpose

Verify the IKE status.

Action

From operational mode, enter the **show security ike sa** command.

```
user@host> show security ike sa
Index  State Initiator cookie          Responder cookie          Mode Remote Address

493333 UP    2001:db8:88b49d915e684c93 2001:db8:fe890b1cac8522b5 Main 2001:db8:3000::2

493334 UP    2001:db8:26e40244ad3d722d 2001:db8:68b4d9f94097d32e Main 2001:db8:5000::2
```

Meaning

The `show security ike sa` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spokes.

Verifying IPsec Status

Purpose

Verify the IPsec status.

Action

From operational mode, enter the **show security ipsec sa** command.

```
user@host> show security ipsec sa
Total active tunnels: 2
  ID        Algorithm      SPI  Life:sec/kb    Mon    lsys Port Gateway
  >67108885 ESP:aes-gcm-256/None fdef4dab 2918/ unlim - root 500  2001:db8:3000::2
  >67108885 ESP:aes-gcm-256/None e785dadc 2918/ unlim - root 500  2001:db8:3000::2
  >67108887 ESP:aes-gcm-256/None 34a787af 2971/ unlim - root 500  2001:db8:5000::2
  >67108887 ESP:aes-gcm-256/None cf57007f 2971/ unlim - root 500  2001:db8:5000::2
```

Meaning

The `show security ipsec sa` command lists all active IKE Phase 2 SAs. If no SAs are listed, there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spokes.

Verifying IPsec Next-Hop Tunnels

Purpose

Verify the IPsec next-hop tunnels.

Action

From operational mode, enter the **show security ipsec next-hop-tunnels** command.

```
user@host> show security ipsec next-hop-tunnels
Next-hop gateway              interface  IPSec VPN name  Flag  IKE-
ID                                XAUTH username

2001:db8:9000::2              st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available

2001:db8:9000::3              st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available

2001:db8::5668:ad10:fcd8:163c st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE1 Not-Available

2001:db8::5668:ad10:fcd8:18a1 st0.1      IPSEC_VPNA_1    Auto  C=US, DC=example.net, ST=CA,
L=Sunnyvale, O=example, OU=SLT, CN=SPOKE2 Not-Available
```

Meaning

The next-hop gateways are the IP addresses for the `st0` interfaces of the spokes. The next hop should be associated with the correct IPsec VPN name.

Verifying OSPFv3

Purpose

Verify that OSPFv3 references the IP addresses for the `st0` interfaces of the spokes.

Action

From operational mode, enter the **show ospf3 neighbor detail** command.

Hub:

```
user@host> show ospf3 neighbor detail
ID                        Interface   State  Pri  Dead
2001:db8:7000:2   st0.1       Full   128   -
  Neighbor-address 2001:db8::5668:ad10:fcd8:18a1
  Area 0.0.0.0, opt 0x33, OSPF3-Intf-Index 2
  DR-ID 0.0.0.0, BDR-ID 0.0.0.0
  Up 00:01:35, adjacent 00:01:31 Hello suppressed 00:01:31 ago
2001:db8:7000:3  st0.1                  Full      128      -
  Neighbor-address 2001:db8::5668:ad10:fcd8:163c
  Area 0.0.0.0, opt 0x33, OSPF3-Intf-Index 2
  DR-ID 0.0.0.0, BDR-ID 0.0.0.0
  Up 00:01:41, adjacent 00:01:37 Hello suppressed 00:01:37 ago
```

Spoke 1:

```
user@host> show ospf3 neighbor detail
ID                        Interface     State     Pri   Dead
2001:db8:7000:1   st0.1         Full      128      -
  Neighbor-address 2001:db8::5668:ad10:fcd8:1946
  Area 0.0.0.0, opt 0x33, OSPF3-Intf-Index 2
  DR-ID 0.0.0.0, BDR-ID 0.0.0.0
  Up 00:05:38, adjacent 00:05:38 Hello suppressed 00:05:34 ago
```

Spoke 2:

```
user@host> show ospf3 neighbor detail
ID                       Interface     State     Pri   Dead
2001:db8:7000:1 st0.1         Full      128      -
```

```
Neighbor-address 2001:db8::5668:ad10:fcd8:1946
Area 0.0.0.0, opt 0x33, OSPF3-Intf-Index 2
DR-ID 0.0.0.0, BDR-ID 0.0.0.0
Up 00:04:44, adjacent 00:04:44 Hello suppressed 00:04:40 ago
```

SEE ALSO

Example: Configuring a Route-Based VPN | 395

Example: Forwarding Traffic Through an AutoVPN Tunnel with Traffic Selectors

IN THIS SECTION

This example shows how to configure traffic selectors, instead of dynamic routing protocols, to forward packets through a VPN tunnel in an AutoVPN deployment. When traffic selectors are configured, the secure tunnel (st0) interface must be in point-to-point mode. Traffic selectors are configured on both the hub and spoke devices.

Requirements

This example uses the following hardware and software components:

- Two SRX Series Firewalls connected and configured in a chassis cluster. The chassis cluster is the AutoVPN hub.
- An SRX Series Firewall configured as an AutoVPN spoke.
- Junos OS Release 12.3X48-D10 or later.

- Digital certificates enrolled in the hub and the spoke devices that allow the devices to authenticate each other.

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates. See "Understanding Local Certificate Requests" on page 52.
- Enroll the digital certificates in each device. See "Example: Loading CA and Local Certificates Manually" on page 63.

Overview

IN THIS SECTION

In this example, traffic selectors are configured on the AutoVPN hub and spoke. Only traffic that conforms to the configured traffic selector is forwarded through the tunnel. On the hub, the traffic selector is configured with the local IP address 192.0.0.0/8 and the remote IP address 172.0.0.0/8. On the spoke, the traffic selector is configured with the local IP address 172.0.0.0/8 and the remote IP address 192.0.0.0/8.

The traffic selector IP addresses configured on the spoke can be a subset of the traffic selector IP addresses configured on the hub. This is known as *traffic selector flexible match*.

Certain Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hubs and spokes must have the same values. Table 107 on page 1245 shows the values used in this example:

Table 107: Phase 1 and Phase 2 Options for AutoVPN Hubs and Spokes with Traffic Selectors

Option	Value
IKE proposal:	
Authentication method	rsa-signatures
Diffie-Hellman (DH) group	group5

Table 107: Phase 1 and Phase 2 Options for AutoVPN Hubs and Spokes with Traffic Selectors *(Continued)*

Option	Value
Authentication algorithm	sha-1
Encryption algorithm	aes-256-cbc
IKE policy:	
Mode	main
Certificate	local-certificate
IKE gateway:	
Dynamic	distinguished name wildcard DC=Common_component
IKE user type	group IKE id
Local identity	distinguished name
Version	v1-only
IPsec proposal:	
Protocol	esp
Authentication algorithm	hmac-sha1-96
Encryption algorithm	aes-192-cbc

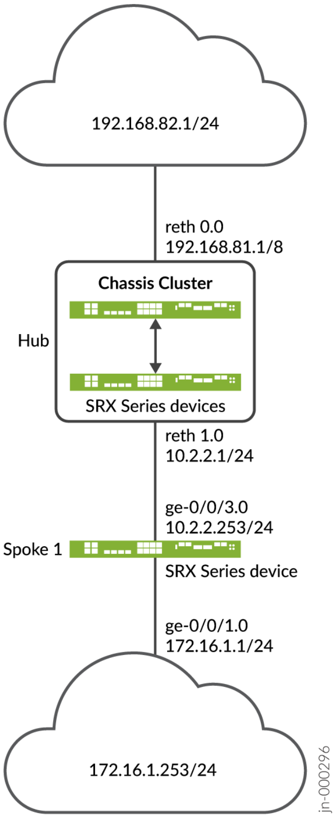
Table 107: Phase 1 and Phase 2 Options for AutoVPN Hubs and Spokes with Traffic Selectors *(Continued)*

Option	Value
Lifetime	3600 seconds 150,000 kilobytes
IPsec policy:	
Perfect Forward Secrecy (PFS) group	group5

Topology

Figure 72 on page 1248 shows the SRX Series Firewalls to be configured for this example.

Figure 72: AutoVPN with Traffic Selectors



Configuration

IN THIS SECTION

Configuring the Hub

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/2 gigether-options redundant-parent reth1
set interfaces ge-0/0/3 gigether-options redundant-parent reth0
set interfaces ge-8/0/2 gigether-options redundant-parent reth1
set interfaces ge-8/0/3 gigether-options redundant-parent reth0
set interfaces lo0 unit 0 family inet address 10.100.1.100/24
set interfaces lo0 redundant-pseudo-interface-options redundancy-group 1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 192.168.81.1/8
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.2.2.1/24
set interfaces st0 unit 1 family inet
set security ike proposal prop_ike authentication-method rsa-signatures
set security ike proposal prop_ike dh-group group5
set security ike proposal prop_ike authentication-algorithm sha1
set security ike proposal prop_ike encryption-algorithm aes-256-cbc
set security ike policy ikepol1 mode main
set security ike policy ikepol1 proposals prop_ike
set security ike policy ikepol1 certificate local-certificate Hub_ID
set security ike gateway HUB_GW ike-policy ikepol1
set security ike gateway HUB_GW dynamic distinguished-name wildcard DC=Domain_component
set security ike gateway HUB_GW dynamic ike-user-type group-ike-id
set security ike gateway HUB_GW local-identity distinguished-name
set security ike gateway HUB_GW external-interface reth1
set security ike gateway HUB_GW version v1-only
set security ipsec proposal prop_ipsec protocol esp
set security ipsec proposal prop_ipsec authentication-algorithm hmac-sha1-96
set security ipsec proposal prop_ipsec encryption-algorithm aes-192-cbc
set security ipsec proposal prop_ipsec lifetime-seconds 3600
set security ipsec proposal prop_ipsec lifetime-kilobytes 150000
set security ipsec policy ipsecpol1 perfect-forward-secrecy keys group5
set security ipsec policy ipsecpol1 proposals prop_ipsec
set security ipsec vpn HUB_VPN bind-interface st0.1
set security ipsec vpn HUB_VPN ike gateway HUB_GW
set security ipsec vpn HUB_VPN ike ipsec-policy ipsecpol1
set security ipsec vpn HUB_VPN traffic-selector ts1 local-ip 192.0.0.0/8
set security ipsec vpn HUB_VPN traffic-selector ts1 remote-ip 172.0.0.0/8
set security pki ca-profile rsa ca-identity rsa
set security pki ca-profile rsa revocation-check disable
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces reth0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces reth1.0
set security policies default-policy permit-all
```

Starting with Junos OS Release 15.1X49-D120, you can configure the CLI option `reject-duplicate-connection` at the [`edit security ike gateway` *`gateway-name`* `dynamic`] hierarchy level to retain an existing tunnel session and reject negotiation requests for a new tunnel with the same IKE ID. By default, an existing tunnel is tear down when a new tunnel with the same IKE ID is established. The `reject-duplicate-connection` option is only supported when `ike-user-type group-ike-id` or `ike-user-type shared-ike-id` is configured for the IKE gateway; the `aaa access-profile` *`profile-name`* configuration is not supported with this option.

Use the CLI option `reject-duplicate-connection` only when you are certain that reestablishment of a new tunnel with the same IKE ID should be rejected.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the hub:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/2 gigether-options redundant-parent reth1
user@host# set ge-0/0/3 gigether-options redundant-parent reth0
user@host# set ge-8/0/2 gigether-options redundant-parent reth1
user@host# set ge-8/0/3 gigether-options redundant-parent reth0
user@host# set lo0 unit 0 family inet address 10.100.1.100/24
user@host# set lo0 redundant-pseudo-interface-options redundancy-group 1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet address 192.168.81.1/8
```

```
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 10.2.2.1/24
user@host# set st0 unit 1 family inet
```

2. Configure Phase 1 options.

```
[edit security ike proposal prop_ike]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy ikepol1]
user@host# set mode main
user@host# set proposals prop_ike
user@host# set certificate local-certificate Hub_ID
[edit security ike gateway HUB_GW]
user@host# set ike-policy ikepol1
user@host# set dynamic distinguished-name wildcard DC=Domain_component
user@host# set dynamic ike-user-type group-ike-id
user@host# set local-identity distinguished-name
user@host# set external-interface reth1
user@host# set version v1-only
```

3. Configure Phase 2 options.

```
[edit security ipsec proposal prop_ipsec]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-192-cbc
user@host# set lifetime-seconds 3600
user@host# set lifetime-kilobytes 150000
[edit security ipsec policy ipsecpol1]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals prop_ipsec
[edit security ipsec HUB_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway HUB_GW
user@host# set ike ipsec-policy ipsecpol1
user@host# set traffic-selector ts1 local-ip 192.0.0.0/8
user@host# set traffic-selector ts1 remote-ip 172.0.0.0/8
```

4. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile rsa ca-identity rsa
user@host# set ca-profile rsa revocation-check disable
```

5. Configure security zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces reth0.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces lo0.0
user@host# set interfaces reth1.0
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike`, `show security ipsec`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/2 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-0/0/3 {
    gigether-options {
        redundant-parent reth0;
    }
```

```
}
lo0 {
    unit 0 {
        family inet {
            address 10.100.1.100/24;
        }
    }
    redundant-pseudo-interface-options {
        redundancy-group 1;
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.168.81.1/8;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 10.2.2.1/24;
        }
    }
}
st0 {
    unit 1 {
        family inet;
    }
}
[edit]
user@host# show security ike
proposal prop_ike {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
```

```
}
policy ikepol1 {
    mode main;
    proposals prop_ike;
    certificate {
        local-certificate Hub_ID;
    }
}
gateway HUB_GW {
    ike-policy ikepol1;
    dynamic distinguished-name wildcard DC=Domain_component;
    dynamic ike-user-type group-ike-id;
    local-identity distinguished-name;
    external-interface reth1;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal prop_ipsec {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-192-cbc;
    lifetime-seconds 3600;
    lifetime-kilobytes 150000;
}
policy ipsecpol1 {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals prop_ipsec;
}
vpn HUB_VPN {
    bind-interface st0.1;
    ike {
        gateway HUB_GW;
        ipsec-policy ipsecpol1;
    }
    traffic-selector ts1 {
        local-ip 192.0.0.0/8;
        remote-ip 172.0.0.0/8;
    }
}
[edit]
```

```
user@host# show security pki
ca-profile rsa {
    ca-identity rsa;
    revocation-check {
        disable;
    }
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.1;
        reth0.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        lo0.0;
        reth1.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Spoke

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24
set interfaces ge-0/0/3 unit 0 family inet address 10.2.2.253/24
set interfaces st0 unit 1 family inet
set security ike proposal prop_ike authentication-method rsa-signatures
set security ike proposal prop_ike dh-group group5
set security ike proposal prop_ike authentication-algorithm sha1
set security ike proposal prop_ike encryption-algorithm aes-256-cbc
set security ike policy ikepol1 mode main
set security ike policy ikepol1 proposals prop_ike
set security ike policy ikepol1 certificate local-certificate Spoke1_ID
set security ike gateway SPOKE_GW ike-policy ikepol1
set security ike gateway SPOKE_GW address 10.2.2.1
set security ike gateway SPOKE_GW local-identity distinguished-name
set security ike gateway SPOKE_GW remote-identity distinguished-name container
DC=Domain_component
set security ike gateway SPOKE_GW external-interface ge-0/0/3.0
set security ike gateway SPOKE_GW version v1-only
set security ipsec proposal prop_ipsec protocol esp
set security ipsec proposal prop_ipsec authentication-algorithm hmac-sha1-96
set security ipsec proposal prop_ipsec encryption-algorithm aes-192-cbc
set security ipsec proposal prop_ipsec lifetime-seconds 3600
set security ipsec proposal prop_ipsec lifetime-kilobytes 150000
set security ipsec policy ipsecpol1 perfect-forward-secrecy keys group5
set security ipsec policy ipsecpol1 proposals prop_ipsec
set security ipsec vpn SPOKE_VPN bind-interface st0.1
set security ipsec vpn SPOKE_VPN ike gateway SPOKE_GW
set security ipsec vpn SPOKE_VPN ike ipsec-policy ipsecpol1
set security ipsec vpn SPOKE_VPN traffic-selector ts1 local-ip 172.0.0.0/8
set security ipsec vpn SPOKE_VPN traffic-selector ts1 remote-ip 192.0.0.0/8
set security ipsec vpn SPOKE_VPN establish-tunnels immediately
set security pki ca-profile rsa ca-identity rsa
set security pki ca-profile rsa revocation-check disable
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces ge-0/0/3.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure the hub:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 172.16.1.1/24
user@host# set ge-0/0/3 unit 0 family inet address 10.2.2.253/24
user@host# set st0 unit 1 family inet
```

2. Configure Phase 1 options.

```
[edit security ike proposal prop_ike]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy ikepol1]
user@host# set mode main
user@host# set proposals prop_ike
user@host# set certificate local-certificate Spoke1_ID
[edit security ike gateway SPOKE_GW]
user@host# set ike-policy ikepol1
user@host# set address 10.2.2.1
user@host# set local-identity distinguished-name
user@host# set remote-identity distinguished-name container DC=Domain_component
```

```
user@host# set external-interface ge-0/0/3.0
user@host# set version v1-only
```

3. Configure Phase 2 options.

```
[edit security ipsec proposal prop_ipsec]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-192-cbc
user@host# set lifetime-seconds 3600
user@host# set lifetime-kilobytes 150000
[edit security ipsec policy ipsecpol1]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals prop_ipsec
[edit security ipsec SPOKE_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway SPOKE_GW
user@host# set ike ipsec-policy ipsecpol1
user@host# set traffic-selector ts1 local-ip 172.0.0.0/8
user@host# set traffic-selector ts1 remote-ip 192.0.0.0/8
user@host# set establish-tunnels immediately
```

4. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile rsa ca-identity rsa
user@host# set ca-profile rsa revocation-check disable
```

5. Configure security zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/3.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces ge-0/0/1.0
```

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security ike`, `show security ipsec`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 172.16.1.1/24;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 10.2.2.253/24;
        }
    }
}
st0 {
    unit 1 {
        family inet;
    }
}
[edit]
user@host# show security ike
proposal prop_ike {
    authentication-method rsa-signatures;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
policy ikepol1 {
    mode main;
    proposals prop_ike;
    certificate {
        local-certificate Spoke1_ID;
    }
}
gateway SPOKE_GW {
    ike-policy ikepol1;
    address 10.2.2.1;
    local-identity distinguished-name;
    remote-identity distinguished-name container DC=Domain_component;
    external-interface ge-0/0/3.0;
    version v1-only;
}
[edit]
user@host# show security ipsec
proposal prop_ipsec {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-192-cbc;
    lifetime-seconds 3600;
    lifetime-kilobytes 150000;
}
policy ipsecpol1 {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals prop_ipsec;
}
vpn SPOKE_VPN {
    bind-interface st0.1;
    ike {
        gateway SPOKE_GW;
        ipsec-policy ipsecpol1;
    }
    traffic-selector ts1 {
        local-ip 172.0.0.0/8;
        remote-ip 192.0.0.0/8;
    }
    establish-tunnels immediately;
}
[edit]
user@host# show security pki
ca-profile rsa {
    ca-identity rsa;
    revocation-check {
        disable;
    }
}
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.1;
        ge-0/0/3.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
[edit]
user@host# show security policies
default-policy {
    permit-all;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Tunnels

Purpose

Verify that tunnels are established between the AutoVPN hub and spoke.

Action

From operational mode, enter the `show security ike security-associations` and `show security ipsec security-associations` commands on the hub.

```
user@host> show security ike security-associations
node0:
--------------------------------------------------------------------------
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
1350248074 UP  d195bce6ccfcf9af  8f1569c6592c8408  Main           10.2.2.253

user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <77594650 ESP:aes-cbc-192/sha1 ac97cb1 2799/  150000 - root 500 10.2.2.253
  >77594650 ESP:aes-cbc-192/sha1 828dc013 2798/  150000 - root 500 10.2.2.253

user@host> show security ipsec security-associations detail
node0:
--------------------------------------------------------------------------
```

```
ID: 77594650 Virtual-system: root, VPN Name: HUB_VPN
  Local Gateway: 10.2.2.1, Remote Gateway: 10.2.2.253
  Traffic Selector Name: ts1
  Local Identity: ipv4(192.0.0.0-192.255.255.255)
  Remote Identity: ipv4(172.0.0.0-172.255.255.255)
  Version: IKEv1
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 2, Fail#: 0, Def-Del#: 0 Flag: 0x24608b29
  Tunnel events:
    Tue Dec 30 2014 11:30:21 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Dec 30 2014 11:30:20 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Dec 30 2014 11:30:20 -0800: IKE SA negotiation successfully completed (3 times)
  Location: FPC 5, PIC 0, KMD-Instance 1
  Direction: inbound, SPI: ac97cb1, AUX-SPI: 0
    Hard lifetime: Expires in 2796 seconds
    Lifesize Remaining:  150000 kilobytes
    Soft lifetime: Expires in 2211 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Location: FPC 5, PIC 0, KMD-Instance 1
  Direction: outbound, SPI: 828dc013, AUX-SPI: 0
    Hard lifetime: Expires in 2796 seconds
    Lifesize Remaining:  150000 kilobytes
    Soft lifetime: Expires in 2211 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

From operational mode, enter the `show security ike security-associations` and `show security ipsec security-associations` commands on the spoke.

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
276505646 UP   d195bce6ccfcf9af  8f1569c6592c8408  Main           10.2.2.1

user@host> show security ipsec security-associations
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <69206018 ESP:aes-cbc-192/sha1 828dc013 2993/  150000 - root 500 10.2.2.1
  >69206018 ESP:aes-cbc-192/sha1 ac97cb1 2993/  150000 - root 500 10.2.2.1
```

```
user@host> show security ipsec security-associations detail
ID: 69206018 Virtual-system: root, VPN Name: SPOKE_VPN
  Local Gateway: 10.2.2.253, Remote Gateway: 10.2.2.1
  Traffic Selector Name: ts1
  Local Identity: ipv4(172.0.0.0-172.255.255.255)
  Remote Identity: ipv4(192.0.0.0-192.255.255.255)
  Version: IKEv1
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x2c608b29
  Tunnel events:
    Tue Dec 30 2014 11:30:20 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Dec 30 2014 11:30:20 -0800: IKE SA negotiation successfully completed (1 times)
    Tue Dec 30 2014 11:26:11 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
  Location: FPC 1, PIC 0, KMD-Instance 1
  Direction: inbound, SPI: 828dc013, AUX-SPI: 0
    Hard lifetime: Expires in 2991 seconds
    Lifesize Remaining:  150000 kilobytes
    Soft lifetime: Expires in 2369 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Location: FPC 1, PIC 0, KMD-Instance 1
  Direction: outbound, SPI: ac97cb1, AUX-SPI: 0
    Hard lifetime: Expires in 2991 seconds
    Lifesize Remaining:  150000 kilobytes
    Soft lifetime: Expires in 2369 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (192 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. The hub shows one active tunnel to the spoke while the spoke shows one active tunnel to the hub.

If no SAs are listed for IKE Phase 1, then there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and spoke.

If no SAs are listed for IKE Phase 2, then there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and spoke.

Verifying Traffic Selectors

Purpose

Verify the traffic selectors.

Action

From operational mode, enter the `show security ipsec traffic-selector interface-name st0.1` command on the hub.

```
user@host> show security ipsec traffic-selector interface-name st0.1
node0:
--------------------------------------------------------------------------
 Source IP                         Destination IP                   Interface   Tunnel-id    IKE-
ID
 192.0.0.0-192.255.255.255           172.0.0.0-172.255.255.255          st0.1       77594650
DC=Domain_component, CN=Spoke1_ID, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
```

From operational mode, enter the `show security ipsec traffic-selector interface-name st0.1` command on the spoke.

```
user@host> show security ipsec traffic-selector interface-name st0.1
 Source IP                         Destination IP                   Interface   Tunnel-id    IKE-
ID
 172.0.0.0-172.255.255.255           192.0.0.0-192.255.255.255          st0.1       69206018
DC=Domain_component, CN=Hub_ID, OU=Sales, O=XYZ, L=Sunnyvale, ST=CA, C=US
```

Meaning

A traffic selector is an agreement between IKE peers to permit traffic through a tunnel if the traffic matches a specified pair of local and remote addresses. Only traffic that conforms to a traffic selector is permitted through an SA. Traffic selectors are negotiated between the initiator and the responder (the SRX Series hub).

SEE ALSO

Understanding Traffic Selectors in Route-Based VPNs | 524

Example: Ensuring VPN Tunnel Availability with AutoVPN and Traffic Selectors

IN THIS SECTION

Georedundancy is the deployment of multiple geographically distant sites so that traffic can continue to flow over a provider network even if there is a power outage, a natural disaster, or other catastrophic event that affects a site. In a mobile provider network, multiple Evolved Node B (eNodeB) devices can be connected to the core network through georedundant IPsec VPN gateways on SRX Series Firewalls. The alternate routes to the eNodeB devices are distributed to the core network using a dynamic routing protocol.

This example configures AutoVPN hubs with multiple traffic selectors on SRX Series Firewalls to ensure that there are georedundant IPsec VPN gateways to eNodeB devices. Auto route insertion (ARI) is used to automatically insert routes toward the eNodeB devices in the routing tables on the hubs. ARI routes are then distributed to the provider's core network through BGP.

Requirements

This example uses the following hardware and software components:

- Two SRX Series Firewalls connected and configured in a chassis cluster. The chassis cluster is AutoVPN hub A.
- An SRX Series Firewall configured as AutoVPN hub B.
- Junos OS Release 12.3X48-D10 or later.
- eNodeB devices that can establish IPsec VPN tunnels with AutoVPN hubs. eNodeB devices are third-party network equipment providers that initiate a VPN tunnel with AutoVPN hubs.

- Digital certificates enrolled in the hubs and the eNodeB devices that allow the devices to authenticate each other.

Before you begin:

- Obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates. See "Understanding Local Certificate Requests" on page 52.
- Enroll the digital certificates in each device. See "Example: Loading CA and Local Certificates Manually" on page 63.

This example uses the BGP dynamic routing protocol to advertise routes toward the eNodeB devices to the core network.

Overview

IN THIS SECTION

In this example, two AutoVPN hubs are configured with multiple traffic selectors on SRX Series Firewalls to provide georedundant IPsec VPN gateways to eNodeB devices. ARI automatically inserts routes to the eNodeB devices in the routing tables on the hubs. ARI routes are then distributed to the provider's core network through BGP.

Certain Phase 1 and Phase 2 IKE tunnel options configured on the AutoVPN hubs and eNodeB devices must have the same values. Table 108 on page 1267 shows the values used in this example:

Table 108: Phase 1 and Phase 2 Options for Georedundant AutoVPN Hubs

Option	Value
IKE proposal:	
Authentication method	rsa-signatures
Diffie-Hellman (DH) group	group5

Table 108: Phase 1 and Phase 2 Options for Georedundant AutoVPN Hubs *(Continued)*

Option	Value
Authentication algorithm	sha-1
Encryption algorithm	aes-256-cbc
IKE policy:	
Certificate	local-certificate
IKE gateway:	
Dynamic	distinguished name wildcard DC=Common_component
IKE user type	group IKE id
Dead peer detection	probe-idle-tunnel
Local identity	distinguished name
Version	v2-only
IPsec proposal:	
Protocol	esp
Authentication algorithm	hmac-sha1-96
Encryption algorithm	aes-256-cbc
IPsec policy:	

Table 108: Phase 1 and Phase 2 Options for Georedundant AutoVPN Hubs *(Continued)*

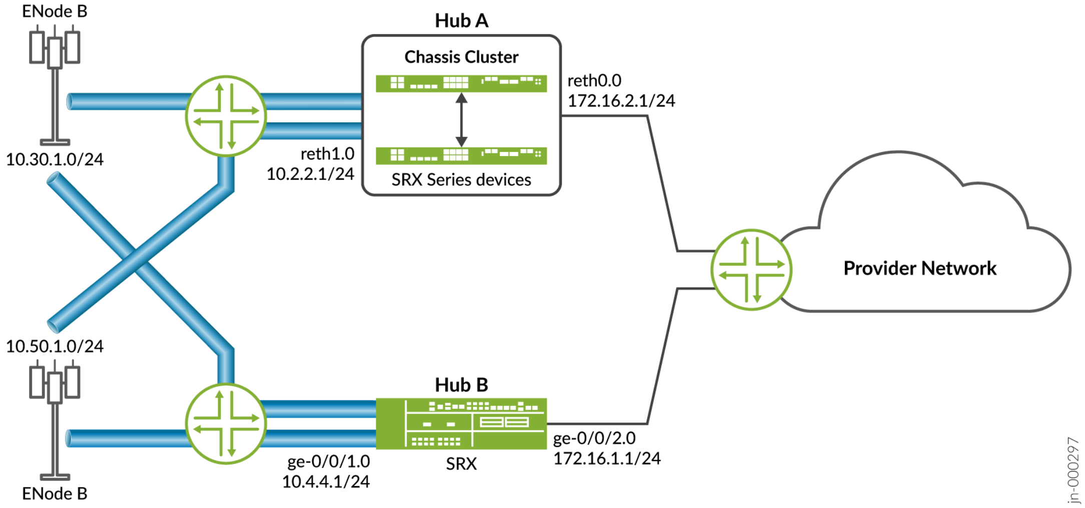
Option	Value
Perfect Forward Secrecy (PFS) group	group5

In this example, the default security policy that permits all traffic is used for all devices. More restrictive security policies should be configured for production environments. See *Security Policies Overview*. For simplicity, the configuration on the SRX Series Firewalls allows all types of inbound traffic; this configuration is not recommended for production deployments.

Topology

Figure 73 on page 1269 shows the SRX Series Firewalls to be configured for this example.

Figure 73: Georedundant IPsec VPN Gateways to eNodeB Devices



Configuration

IN THIS SECTION

Configuring Hub A

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter commit from configuration mode.

```
set interfaces ge-0/0/2 gigether-options redundant-parent reth1
set interfaces ge-0/0/3 gigether-options redundant-parent reth0
set interfaces ge-8/0/2 gigether-options redundant-parent reth1
set interfaces ge-8/0/3 gigether-options redundant-parent reth0
set interfaces lo0 unit 0 family inet address 10.100.1.100/24
set interfaces lo0 redundant-pseudo-interface-options redundancy-group 1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 172.16.2.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.2.2.1/24
set interfaces st0 unit 1 family inet
set security ike proposal prop_ike authentication-method rsa-signatures
set security ike proposal prop_ike dh-group group5
set security ike proposal prop_ike authentication-algorithm sha1
set security ike proposal prop_ike encryption-algorithm aes-256-cbc
set security ike policy ph1_ike_policy proposals prop_ike
set security ike policy ph1_ike_policy certificate local-certificate HubA_certificate
set security ike gateway HUB_GW ike-policy ph1_ike_policy
set security ike gateway HUB_GW dynamic distinguished-name wildcard DC=Common_component
set security ike gateway HUB_GW dynamic ike-user-type group-ike-id
set security ike gateway HUB_GW dead-peer-detection probe-idle-tunnel
set security ike gateway HUB_GW local-identity distinguished-name
set security ike gateway HUB_GW external-interface reth1
set security ike gateway HUB_GW version v2-only
set security ipsec proposal prop_ipsec protocol esp
set security ipsec proposal prop_ipsec authentication-algorithm hmac-sha1-96
set security ipsec proposal prop_ipsec encryption-algorithm aes-256-cbc
set security ipsec policy ph2_ipsec_policy perfect-forward-secrecy keys group5
set security ipsec policy ph2_ipsec_policy proposals prop_ipsec
set security ipsec vpn HUB_VPN bind-interface st0.1
set security ipsec vpn HUB_VPN ike gateway HUB_GW
set security ipsec vpn HUB_VPN ike ipsec-policy ph2_ipsec_policy
set security ipsec vpn HUB_VPN traffic-selector ts1 local-ip 172.16.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts1 remote-ip 10.50.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts2 local-ip 172.16.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts2 remote-ip 10.30.0.0/16
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 172.16.2.1
set protocols bgp group internal-peers export inject_ts1_routes
set protocols bgp group internal-peers export inject_ts2_routes
set protocols bgp group internal-peers export inject_up_routes
set protocols bgp group internal-peers neighbor 172.16.2.4
set policy-options policy-statement inject_ts1_routes term cp_allow from protocol static
set policy-options policy-statement inject_ts1_routes term cp_allow from route-filter
10.30.1.0/24 orlonger
set policy-options policy-statement inject_ts1_routes term cp_allow from route-filter
10.30.1.0/24 orlonger
set policy-options policy-statement inject_ts1_routes term cp_allow then next-hop self
set policy-options policy-statement inject_ts1_routes term cp_allow then accept
set policy-options policy-statement inject_ts2_routes term mp_allow from protocol static
set policy-options policy-statement inject_ts2_routes term mp_allow from route-filter
10.50.1.0/24 orlonger
set policy-options policy-statement inject_ts2_routes term mp_net_allow from route-filter
10.50.2.0/24 orlonger
set policy-options policy-statement inject_ts2_routes term mp_net_allow then next-hop self
set policy-options policy-statement inject_ts2_routes term mp_net_allow then accept
set policy-options policy-statement inject_up_routes term up_allow from protocol static
set policy-options policy-statement inject_up_routes term up_allow from route-filter
172.16.1.0/24 orlonger
set policy-options policy-statement inject_up_routes term up_allow from route-filter
172.16.2.0/24 orlonger
set policy-options policy-statement inject_up_routes term up_allow then next-hop self
set policy-options policy-statement inject_up_routes term up_allow then accept
set security pki ca-profile csa ca-identity csa
set security pki ca-profile csa revocation-check disable
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces reth0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces reth1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure hub A:

1. Configure interfaces.

   ```
   [edit interfaces]
   user@host# set ge-0/0/2 gigether-options redundant-parent reth1
   user@host# set ge-0/0/3 gigether-options redundant-parent reth0
   user@host# set ge-8/0/2 gigether-options redundant-parent reth1
   user@host# set ge-8/0/3 gigether-options redundant-parent reth0
   user@host# set lo0 unit 0 family inet address 10.100.1.100/24
   user@host# set lo0 redundant-pseudo-interface-options redundancy-group 1
   user@host# set reth0 redundant-ether-options redundancy-group 1
   user@host# set reth0 unit 0 family inet address 172.16.2.1/24
   user@host# set reth1 redundant-ether-options redundancy-group 1
   user@host# set reth1 unit 0 family inet address 10.2.2.1/24
   user@host# set st0 unit 1 family inet
   ```

2. Configure Phase 1 options.

   ```
   [edit security ike proposal prop_ike]
   user@host# set authentication-method rsa-signatures
   user@host# set dh-group group5
   user@host# set authentication-algorithm sha1
   user@host# set encryption-algorithm aes-256-cbc
   [edit security ike policy ph1_ike_policy]
   user@host# set proposals prop_ike
   user@host# set certificate local-certificate HubA_certificate
   [edit security ike gateway HUB_GW]
   user@host# set ike-policy ph1_ike_policy
   user@host# set dynamic distinguished-name wildcard DC=Common_component
   user@host# set dynamic ike-user-type group-ike-id
   ```

```
user@host# set dead-peer-detection probe-idle-tunnel
user@host# set local-identity distinguished-name
user@host# set external-interface reth1
user@host# set version v2-only
```

3. Configure Phase 2 options.

```
[edit security ipsec proposal prop_ipsec]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy ph2_ipsec_policy]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals prop_ipsec
[edit security ipsec vpn HUB_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway HUB_GW
user@host# set ike ipsec-policy ph2_ipsec_policy
user@host# set traffic-selector ts1 local-ip 172.16.0.0/16
user@host# set traffic-selector ts1 remote-ip 10.50.0.0/16
user@host# set traffic-selector ts2 local-ip 172.16.0.0/16
user@host# set traffic-selector ts2 remote-ip 10.30.0.0/16
```

4. Configure the BGP routing protocol.

```
[edit protocols bgp group internal-peers]
user@host# set type internal
user@host# set local-address 172.16.2.1
user@host# set export inject_ts1_routes
user@host# set export inject_ts2_routes
user@host# set export inject_up_routes
user@host# set neighbor 172.16.2.4
```

5. Configure routing options.

```
[edit policy-options policy-statement inject_ts1_routes]
user@host# set term cp_allow from protocol  static
user@host# set term cp_allow from route-filter 10.30.2.0/24 orlonger
user@host# set term cp_allow from route-filter 10.30.1.0/24 orlonger
user@host# set term cp_allow then next-hop  self
```

```
user@host# set term cp_allow then accept
[edit policy-options policy-statement inject_ts2_routes]
user@host# set term mp_allow from protocol  static
user@host# set term mp_allow from route-filter 10.50.1.0/24 orlonger
user@host# set term mp_allow from  route-filter 10.50.2.0/24 orlonger
user@host# set term mp_allow then next-hop  self
user@host# set term mp_allow then accept
[edit policy-options policy-statement inject_up_routes]
user@host# set term up_allow from protocol  static
user@host# set term up_allow from route-filter 172.16.1.0/24 orlonger
user@host# set term up_allow from  route-filter 172.16.2.0/24 orlonger
user@host# set term up_allow then next-hop  self
user@host# set term up_allow then accept
```

6. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile csa ca-identity csa
user@host# set ca-profile csa revocation-check disable
```

7. Configure security zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces reth0.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces lo0.0
user@host# set interfaces reth1.0
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` `show security ike`, `show security ipsec`, `show protocols bgp`, `show policy-options`, `show security pki`, `show security zones`, and `show security` policies commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
    ge-0/0/2 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-0/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    ge-8/0/2 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-8/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.100.1.100/24;
            }
        }
        redundant-pseudo-interface-options {
            redundancy-group 1;
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 172.16.2.1/16;
            }
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 10.2.2.1/24;
            }
        }
    }
    st0 {
        unit 1 {
            family inet;
        }
    }
[edit]
user@host# show security ike
    proposal prop_ike {
        authentication-method rsa-signatures;
        dh-group group5;
        authentication-algorithm sha1;
        encryption-algorithm aes-256-cbc;
    }
    policy ph1_ike_policy {
        proposals prop_ike;
        certificate {
            local-certificate HubA_certificate;
        }
    }
    gateway HUB_GW {
        ike-policy ph1_ike_policy;
        dynamic {
            distinguished-name {
                wildcard DC=Common_component;
            }
            ike-user-type group-ike-id;
        }
        dead-peer-detection {
            probe-idle-tunnel;
        }
        local-identity distinguished-name;
        external-interface reth1;
        version v2-only;
    }
[edit]
user@host# show security ipsec
    proposal prop_ipsec {
        protocol esp;
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm aes-256-cbc;
    }
    policy ph2_ipsec_policy {
        perfect-forward-secrecy {
            keys group5;
        }
        proposals prop_ipsec;
    }
    vpn HUB_VPN {
        bind-interface st0.1;
        ike {
            gateway HUB_GW;
            ipsec-policy ph2_ipsec_policy;
        }
        traffic-selector ts1 {
            local-ip 172.16.0.0/16;
            remote-ip 10.50.0.0/16;
        }
        traffic-selector ts2 {
            local-ip 172.16.0.0/16;
            remote-ip 10.30.0.0/16;
        }
    }
[edit]
user@host# show protocols bgp
    group internal-peers {
        type internal;
        local-address 172.16.2.1;
            export [ inject_ts1_routes inject_ts2_routes inject_up_routes ];
        neighbor 172.16.2.4;
    }
[edit]
user@host# show policy-options
policy-statement inject_ts1_routes {
    term cp_allow {
        from {
            protocol static;
            route-filter 10.30.2.0/24 orlonger;
            route-filter 10.30.1.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement inject_ts2_routes {
    term mp_allow {
        from {
            protocol static;
            route-filter 10.50.1.0/24 orlonger;
            route-filter 10.50.2.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement inject_up_routes {
    term up_allow {
        from {
            protocol static;
            route-filter 172.16.1.0/24 orlonger;
            route-filter 172.16.2.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
[edit]
user@host# show security pki
ca-profile csa {
    ca-identity csa;
    revocation-check {
        disable;
    }
}
[edit]
user@host# show security zones
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            st0.1;
            reth0.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            lo0.0;
            reth1.0;
        }
    }
[edit]
user@host# show security policies
    default-policy {
        permit-all;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring Hub B

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 10.4.4.1/24
set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/16
set interfaces lo0 unit 0 family inet address 10.100.1.101/24
set interfaces st0 unit 1 family inet
set security ike proposal prop_ike authentication-method rsa-signatures
set security ike proposal prop_ike dh-group group5
set security ike proposal prop_ike authentication-algorithm sha1
set security ike proposal prop_ike encryption-algorithm aes-256-cbc
set security ike policy ph1_ike_policy proposals prop_ike
set security ike policy ph1_ike_policy certificate local-certificate HubB_certificate
set security ike gateway HUB_GW ike-policy ph1_ike_policy
set security ike gateway HUB_GW dynamic distinguished-name wildcard DC=Common_component
set security ike gateway HUB_GW dynamic ike-user-type group-ike-id
set security ike gateway HUB_GW dead-peer-detection probe-idle-tunnel
set security ike gateway HUB_GW local-identity distinguished-name
set security ike gateway HUB_GW external-interface ge-0/0/1
set security ike gateway HUB_GW version v2-only
set security ipsec proposal prop_ipsec protocol esp
set security ipsec proposal prop_ipsec authentication-algorithm hmac-sha1-96
set security ipsec proposal prop_ipsec encryption-algorithm aes-256-cbc
set security ipsec policy ph2_ipsec_policy perfect-forward-secrecy keys group5
set security ipsec policy ph2_ipsec_policy proposals prop_ipsec
set security ipsec vpn HUB_VPN bind-interface st0.1
set security ipsec vpn HUB_VPN ike gateway HUB_GW
set security ipsec vpn HUB_VPN ike ipsec-policy ph2_ipsec_policy
set security ipsec vpn HUB_VPN traffic-selector ts1 local-ip 172.16.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts1 remote-ip 10.50.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts2 local-ip 172.16.0.0/16
set security ipsec vpn HUB_VPN traffic-selector ts2 remote-ip 10.30.0.0/8
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers local-address 172.16.1.1
set protocols bgp group internal-peers export inject_ts1_routes
set protocols bgp group internal-peers export inject_ts2_routes
set protocols bgp group internal-peers export inject_up_routes
set policy-options policy-statement inject_ts1_routes term cp_allow from protocol static
set policy-options policy-statement inject_ts1_routes term cp_allow from route-filter
10.30.2.0/24 orlonger
set policy-options policy-statement inject_ts1_routes term cp_allow from route-filter
10.30.1.0/24 orlonger
set policy-options policy-statement inject_ts1_routes term cp_allow then next-hop self
set policy-options policy-statement inject_ts1_routes term cp_allow then accept
set policy-options policy-statement inject_ts2_routes term mp_allow from protocol static
set policy-options policy-statement inject_ts2_routes term mp_allow from route-filter
10.50.1.0/24 orlonger
set policy-options policy-statement inject_ts2_routes term mp_net_allow from route-filter
10.50.2.0/24 orlonger
set policy-options policy-statement inject_ts2_routes term mp_net_allow then next-hop self
set policy-options policy-statement inject_ts2_routes term mp_net_allow then accept
set policy-options policy-statement inject_up_routes term up_allow from protocol static
set policy-options policy-statement inject_up_routes term up_allow from route-filter
172.16.1.0/24 orlonger
set policy-options policy-statement inject_up_routes term up_allow from route-filter
172.16.2.0/24 orlonger
set policy-options policy-statement inject_up_routes term up_allow then next-hop self
set policy-options policy-statement inject_up_routes term up_allow then accept
set security pki ca-profile csa ca-identity csa
set security pki ca-profile csa revocation-check disable
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces st0.1
set security zones security-zone trust interfaces ge-0/0/2.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure hub B:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 10.4.4.1/24
user@host# set ge-0/0/2 unit 0 family inet address 172.16.1.1/16
user@host# set lo0 unit 0 family inet address 10.100.1.101/24
user@host# set st0 unit 1 family inet
```

2. Configure Phase 1 options.

```
[edit security ike proposal prop_ike]
user@host# set authentication-method rsa-signatures
user@host# set dh-group group5
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm aes-256-cbc
[edit security ike policy ph1_ike_policy]
user@host# set proposals prop_ike
user@host# set certificate local-certificate HubB_certificate
[edit security ike gateway HUB_GW]
user@host# set ike-policy ph1_ike_policy
user@host# set dynamic distinguished-name wildcard DC=Common_component
user@host# set dynamic ike-user-type group-ike-id
user@host# set dead-peer-detection probe-idle-tunnel
user@host# set local-identity distinguished-name
user@host# set external-interface ge-0/0/1
user@host# set version v2-only
```

3. Configure Phase 2 options.

```
[edit security ipsec proposal prop_ipsec]
user@host# set protocol esp
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm aes-256-cbc
[edit security ipsec policy ph2_ipsec_policy]
user@host# set perfect-forward-secrecy keys group5
user@host# set proposals prop_ipsec
[edit security ipsec vpn HUB_VPN]
user@host# set bind-interface st0.1
user@host# set ike gateway HUB_GW
user@host# set ike ipsec-policy ph2_ipsec_policy
```

```
user@host# set traffic-selector ts1 local-ip 172.16.0.0/16
user@host# set traffic-selector ts1 remote-ip 10.50.0.0/16
user@host# set traffic-selector ts2 local-ip 172.16.0.0/16
user@host# set traffic-selector ts2 remote-ip 10.30.0.0/16
```

4. Configure the BGP routing protocol.

```
[edit protocols bgp group internal-peers]
user@host# set type internal
user@host# set local-address 172.16.1.1
user@host# set export inject_ts1_routes
user@host# set export inject_ts2_routes
user@host# set export inject_up_routes
user@host# set neighbor 172.16.1.2
```

5. Configure routing options.

```
[edit policy-options policy-statement inject_ts1_routes]
user@host# set term cp_allow from protocol  static
user@host# set term cp_allow from route-filter 10.30.2.0/24 orlonger
user@host# set term cp_allow from route-filter 10.30.1.0/24 orlonger
user@host# set term cp_allow then next-hop  self
user@host# set term cp_allow then accept
[edit policy-options policy-statement inject_ts2_routes]
user@host# set term mp_allow from protocol  static
user@host# set term mp_allow from route-filter 10.50.1.0/24 orlonger
user@host# set term mp_allow from route-filter 10.50.2.0/24 orlonger
user@host# set term mp_allow then next-hop  self
user@host# set term mp_allow then accept
[edit policy-options policy-statement inject_up_routes]
user@host# set term up_allow from protocol  static
user@host# set term up_allow from route-filter 172.16.1.0/24 orlonger
user@host# set term up_allow from  route-filter 172.16.2.0/24 orlonger
user@host# set term up_allow then next-hop  self
user@host# set term up_allow then accept
```

6. Configure certificate information.

```
[edit security pki]
user@host# set ca-profile csa ca-identity csa
user@host# set ca-profile csa revocation-check disable
```

7. Configure security zones.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces st0.1
user@host# set interfaces ge-0/0/2.0
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
user@host# set interfaces lo0.0
user@host# set interfaces ge-0/0/1.0
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` `show security ike`, `show security ipsec`, `show protocols bgp`, `show security pki`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.4.4.1/24;
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family inet {
                address 172.16.1.1/16;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.100.1.101/24;
            }
        }
    }
    st0 {
        unit 1 {
            family inet;
        }
    }
[edit]
user@host# show security ike
    proposal prop_ike {
        authentication-method rsa-signatures;
        dh-group group5;
        authentication-algorithm sha1;
        encryption-algorithm aes-256-cbc;
    }
    policy ph1_ike_policy {
        proposals prop_ike;
        certificate {
            local-certificate HubB_certificate;
        }
    }
    gateway HUB_GW {
        ike-policy ph1_ike_policy;
        dynamic {
            distinguished-name {
                wildcard DC=Common_component;
            }
            ike-user-type group-ike-id;
        }
        dead-peer-detection {
            probe-idle-tunnel;
        }
        local-identity distinguished-name;
        external-interface reth1;
        version v2-only;
    }
[edit]
user@host# show security ipsec
    proposal prop_ipsec {
        protocol esp;
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm aes-256-cbc;
    }
    policy ph2_ipsec_policy {
        perfect-forward-secrecy {
            keys group5;
        }
        proposals prop_ipsec;
    }
    vpn HUB_VPN {
        bind-interface st0.1;
        ike {
            gateway HUB_GW;
            ipsec-policy ph2_ipsec_policy;
        }
        traffic-selector ts1 {
            local-ip 172.16.0.0/16;
            remote-ip 10.50.0.0/16;
        }
        traffic-selector ts2 {
            local-ip 172.16.0.0/16;
            remote-ip 10.30.0.0/16;
        }
    }
[edit]
user@host# show protocols bgp
    group internal-peers {
        type internal;
        local-address 172.16.1.1;
            export [ inject_ts1_routes inject_ts2_routes inject_up_routes ];
        neighbor 172.16.1.2;
    }
user@host# show policy-options
policy-statement inject_ts1_routes {
    term cp_allow {
        from {
            protocol static;
            route-filter 10.30.2.0/24 orlonger;
            route-filter 10.30.1.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement inject_ts2_routes {
    term mp_allow {
        from {
            protocol static;
            route-filter 10.50.1.0/24 orlonger;
            route-filter 10.50.2.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
policy-statement inject_up_routes {
    term up_allow {
        from {
            protocol static;
            route-filter 172.16.1.0/24 orlonger;
            route-filter 172.16.2.0/24 orlonger;
        }
        then {
            next-hop self;
            accept;
        }
    }
}
[edit]
user@host# show security pki
ca-profile csa {
    ca-identity csa;
    revocation-check {
        disable;
    }
```

```
}
[edit]
user@host# show security zones
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            st0.1;
            ge-0/0/2.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/1.0;
            lo0.0;
        }
    }
[edit]
user@host# show security policies
    default-policy {
        permit-all;
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the eNodeB (Sample Configuration)

Step-by-Step Procedure

1. The eNodeB configuration in this example is provided for reference. Detailed eNodeB configuration information is beyond the scope of this document. The eNodeB configuration must include the following information:

    - Local certificate (X.509v3) and IKE identity information

    - SRX Series IKE identity information and public IP address

    - Phase 1 and Phase 2 proposals that match the configurations on the SRX Series hubs

Results

The eNodeB devices in this example use strongSwan open source software for IPsec-based VPN connections:

```
config setup
        plutostart=yes
        plutodebug=all
        charondebug="ike 4, cfg 4, chd 4, enc 1"
        charonstart=yes  #ikev2 deamon"
        nat_traversal=yes  #<======= need to enable even no nat_t

conn %default
        ikelifetime=60m
        keylife=45m
        rekeymargin=2m
        keyingtries=4
        mobike=no

conn Hub_A
        keyexchange=ikev2
        authby=pubkey
        ike=aes256-sha-modp1536
        esp=aes256-sha1-modp1536
        leftcert=/usr/local/etc/ipsec.d/certs/fight02Req.pem.Email.crt
        left=10.5.5.1 # self if
        leftsubnet=10.1.1.0/24 # left subnet
        leftid="CN=fight02, DC=Common_component, OU=Dept, O=Company, L=City, ST=CA, C=US " #
self id
```

```
        right=10.2.2.1 # peer if
        rightsubnet=10.1.1.0/24 # peer net for proxy id
        rightid="DC=Domain_component, CN=HubA_certificate, OU=Dept, O=Company, L=City, ST=CA,
C=US " # peer id
        auto=add
        leftfirewall=yes
        dpdaction=restart
        dpddelay=10
        dpdtimeout=120
        rekeyfuzz=10%
        reauth=no

conn Hub_B
        keyexchange=ikev2
        authby=pubkey
        ike=aes256-sha-modp1536
        esp=aes192-sha1-modp1536
        leftcert=/usr/local/etc/ipsec.d/certs/fight02Req.pem.Email.crt
        left=10.5.5.1 # self if
        leftsubnet=10.1.1.0/24 # self net for proxy id
        leftid="CN=fight02, DC=Common_component, OU=Dept, O=Company, L=City, ST=CA, C=US " #
self id
        right=10.4.4.1 # peer if
        rightsubnet=10.1.1.0/24 # peer net for proxy id
        rightid="DC=Domain_component, CN=HubB_certificate, OU=Dept, O=Company, L=City, ST=CA,
C=US " # peer id
        auto=add
        leftfirewall=yes
        dpdaction=restart
        dpddelay=10
        dpdtimeout=120
        rekeyfuzz=10%
        reauth=no
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Tunnels on the AutoVPN Hubs

Purpose

Verify that tunnels are established between the AutoVPN hub and eNodeB devices.

Action

From operational mode, enter the `show security ike security-associations` and `show security ipsec security-associations` commands on the hub.

```
user@host> show security ike security-associations
node0:
--------------------------------------------------------------------------
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
276505706 UP   16d6e53f0866b5cc  ccd8ca944da7b63e  IKEv2          10.5.5.1
1350247532 UP  d5f0cb3a3b18cb92  91269f05527217a0  IKEv2          10.1.1.1

user@host> show security ipsec security-associations
node0:
--------------------------------------------------------------------------
  Total active tunnels: 2
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <77594626 ESP:aes-cbc-192/sha1 a82bbc3 3600/  64 - root 500  10.1.1.1
  >77594626 ESP:aes-cbc-192/sha1 c930a858 3600/  64 - root 500 10.1.1.1
  <69206018 ESP:aes-cbc-192/sha1 2b437fc 3600/  64 - root 500  10.5.5.1
  >69206018 ESP:aes-cbc-192/sha1 c6e02755 3600/  64 - root 500 10.5.5.1
```

Meaning

The `show security ike security-associations` command lists all active IKE Phase 1 SAs. The `show security ipsec security-associations` command lists all active IKE Phase 2 SAs. The hub shows two active tunnels, one to each eNodeB device.

If no SAs are listed for IKE Phase 1, then there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 1 proposal parameters must match on the hub and eNodeB devices.

If no SAs are listed for IKE Phase 2, then there was a problem with Phase 2 establishment. Check the IKE policy parameters and external interface settings in your configuration. Phase 2 proposal parameters must match on the hub and eNodeB devices.

Verifying Traffic Selectors

Purpose

Verify the traffic selectors.

Action

From operational mode, enter the `show security ipsec traffic-selector interface-name st0.1` command.

```
user@host> show security ipsec traffic-selector interface-name st0.1
node0:
---------------------------------------------------------------------------
 Source IP                         Destination IP                    Interface   Tunnel-id    IKE-
ID
 10.1.1.0-10.1.1.255               10.1.1.0-10.1.1.255               st0.1       69206018
DC=Common_component, CN=enodebA, OU=Dept, O=Company, L=City, ST=CA, C=US
 10.1.1.0-10.1.1.255               10.1.1.0-10.1.1.255               st0.1       77594626
DC=Common_component, CN=enodebB, OU=Dept, O=Company, L=City, ST=CA, C=US
```

Meaning

A traffic selector is an agreement between IKE peers to permit traffic through a tunnel if the traffic matches a specified pair of local and remote addresses. Only traffic that conforms to a traffic selector is permitted through an SA. Traffic selectors are negotiated between the initiator and the responder (the SRX Series hub).

Verifying ARI Routes

Purpose

Verify that the ARI routes are added to the routing table.

Action

From operational mode, enter the show route command.

```
user@host> show route
inet.0: 23 destinations, 23 routes (22 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.0.0/16         *[Static/5] 02:57:57
                    > to 2.2.2.253 via reth1.0
10.2.2.0/24         *[Direct/0] 02:58:43
                    > via reth1.0
10.2.2.1/32         *[Local/0] 02:59:25
                      Local via reth1.0
10.5.0.0/16         *[Static/5] 02:57:57
                    > to 2.2.2.253 via reth1.0
10.157.64.0/19     *[Direct/0] 21:54:52
                    > via fxp0.0
10.157.75.117/32   *[Local/0] 21:54:52
                      Local via fxp0.0
10.254.75.117/32   *[Direct/0] 21:54:52
                    > via lo0.0
10.30.1.0/24        *[ARI-TS/5] 02:28:10        [ARI route added based on TSi]
                    > via st0.1
10.50.1.0/24        *[ARI-TS/5] 02:28:26
                    > via st0.1
10.80.0.0/16         *[Direct/0] 02:57:57
                    > via reth0.0
10.80.1.1/32        *[Local/0] 02:57:57
                      Local via reth0.0
10.100.1.0/24     *[Direct/0] 02:57:57
                    > via lo0.0
10.100.1.100/32   *[Local/0] 02:57:57
                      Local via lo0.0
10.102.1.0/24     *[Static/5] 02:57:57
                    > to 10.2.2.253 via reth1.0
10.104.1.0/24     *[Static/5] 02:57:57
                    > to 10.2.2.253 via reth1.0
172.16.0.0/12      *[Static/5] 21:54:52
```

Meaning

Auto route insertion (ARI) automatically inserts a static route for the remote network and hosts protected by a remote tunnel endpoint. A route is created based on the remote IP address configured in the traffic selector. In the case of traffic selectors, the configured remote address is inserted as a route in the routing instance associated with the st0 interface that is bound to the VPN.

Static routes to the eNodeB destinations 10.30.1.0/24 and 10.50.1.0/24 are added to the routing table on the SRX Series hub. These routes are reachable through the st0.1 interface.

SEE ALSO

Understanding Traffic Selectors in Route-Based VPNs | 524

Example: Configuring AutoVPN with Pre-Shared Key

IN THIS SECTION

This example shows how to configure different IKE preshared key used by the VPN gateway to authenticate the remote peer. Similarly, to configure same IKE preshared key used by the VPN gateway to authenticate the remote peer.

Requirements

This example uses the following hardware and software components:

- MX240, MX480, and MX960 with MX-SPC3 and Junos OS Release 21.1R1 that support AutoVPN
- or SRX5000 line with SPC3 and Junos OS Release 21.2R1 that support AutoVPN
- or vSRX Virtual Firewall running iked and Junos OS Release 21.2R1 that support AutoVPN

Configure different IKE preshared key

To configure different IKE preshared key that the VPN gateway uses to authenticate the remote peer, perform these tasks.

1. Configure the seeded preshared for IKE policy in the device with AutoVPN hub.

   ```
    [edit]
   user@host# set security ike policy IKE_POL seeded-pre-shared-key ascii-text ascii-text
   ```

   or

   ```
   user@host# set security ike policy IKE_POL seeded-pre-shared-key hexadecimal hexadecimal
   ```

   For example:

   ```
   user@host# set security ike policy IKE_POL seeded-pre-shared-key ascii-text
   ThisIsMySecretPreSharedkey
   ```

   or

   ```
   user@host# set security ike policy IKE_POL seeded-pre-shared-key hexadecimal
   5468697349734d79536563726563745072655368617265646b6579
   ```

2. Display the `pre-shared key` for remote peer using gateway name and user-id.

   ```
    [edit]
   user@host> show security ike pre-shared-key gateway gateway-name user-id user-id
   ```

   For example:

   ```
   user@host> show security ike pre-shared-key gateway-name HUB_GW user-id user1@juniper.net
   ```

   ```
   Pre-shared key: 79e4ea39f5c06834a3c4c031e37c6de24d46798a
   ```

3. Configure the generated PSK ("79e4ea39f5c06834a3c4c031e37c6de24d46798a" in "step 2" on page 1295) in the ike policy on the remote peer device.

   ```
    [edit]
   user@peer# set security ike policy IKE_POL pre-shared-key ascii-text generated-psk
   ```

   For example:

   ```
   user@peer# set security ike policy IKE_POL pre-shared-key ascii-text
   79e4ea39f5c06834a3c4c031e37c6de24d46798a
   ```

4. (Optional) To bypass the IKE ID validation and allow all IKE ID types, configure `general-ikeid` configuration statement under the [edit security ike gateway *gateway_name* dynamic] hierarchy level in the gateway.

   ```
    [edit]
   user@host# set security ike gateway HUB_GW dynamic general-ikeid
   ```

Result

From the configuration mode, confirm your configuration by entering the show security command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host> show security
    ike {
        proposal IKE_PROP {
            authentication-method pre-shared-keys;
            dh-group group14;
            authentication-algorithm sha-256;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 750;
        }
        policy IKE_POL {
          proposals IKE_PROP;
          seeded-pre-shared-key ascii-text "$9$zoDln9pIEyWLN0BLNdboaFn/C0BRhSeM8"; ##SECRET-DATA
        }
        gateway HUB_GW {
            ike-policy IKE_POL;
```

```
            dynamic {
                general-ikeid;
                ike-user-type group-ike-id;
            }
            local-identity hostname hub.juniper.net;
            external-interface lo0.0;
            local-address 11.0.0.1;
            version v2-only;
        }
    }
```

Configure same IKE preshared key

To configure same IKE preshared key that the VPN gateway uses to authenticate the remote peer, perform these tasks.

1. Configure the common `pre-shared-key` for ike policy in the device with AutoVPN hub.

   ```
    [edit]
   user@host# set security ike policy IKE_POL pre-shared-key ascii-text ascii text
   ```

   For example:

   ```
   user@host# # set security ike policy IKE_POL pre-shared-key ascii-text
   ThisIsMySecretPreSharedkey
   ```

2. Configure the common `pre-shared-key` on the ike policy for remote peer device.

   ```
    [edit]
   user@peer# set security ike policy IKE_POL pre-shared-key ascii-text ascii text
   ```

   For example:

   ```
   user@peer# set security ike policy IKE_POL pre-shared-key ascii-text
   ThisIsMySecretPreSharedkey
   ```

3. (Optional) To bypass the IKE ID validation and allow all IKE ID types, configure `general-ikeid` configuration statement under the [edit security ike gateway *gateway_name* dynamic] hierarchy level in the gateway.

   ```
   [edit]
   user@host# set security ike gateway HUB_GW dynamic general-ikeid
   ```

Result

From the configuration mode, confirm your configuration by entering the show security command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host> show security
    ike {
        proposal IKE_PROP {
            authentication-method pre-shared-keys;
            dh-group group14;
            authentication-algorithm sha-256;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 750;
        }
        policy IKE_POL {
            proposals IKE_PROP;
            pre-shared-key ascii-text "$9$wo2oGk.569pDi9p0BSys24"; ## SECRET-DATA
        }
        gateway HUB_GW {
            ike-policy IKE_POL;
            dynamic {
                general-ikeid;
                ike-user-type group-ike-id;
            }
            local-identity user-at-hostname user1@juniper.net;
            external-interface lo0;
            local-address 11.0.0.1;
            version v2-only;
        }
    }
```

Release History Table

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, IPv6 address is supported on AutoVPN.
17.4R1	Starting with Junos OS Release 17.4R1, AutoVPN networks that use secure tunnel interfaces in point-to-point mode support IPv6 addresses for traffic selectors and for IKE peers.
15.1X49-D120	Starting with Junos OS Release 15.1X49-D120, you can configure the CLI option `reject-duplicate-connection` at the [`edit security ike gateway` *`gateway-name`* `dynamic`] hierarchy level to retain an existing tunnel session and reject negotiation requests for a new tunnel with the same IKE ID.

RELATED DOCUMENTATION

Monitoring VPN Traffic | **1351**

13

CHAPTER

Remote Access VPN

Juniper Secure Connect

IN THIS SECTION

Read this topic to get an overview about Juniper Secure Connect solution.

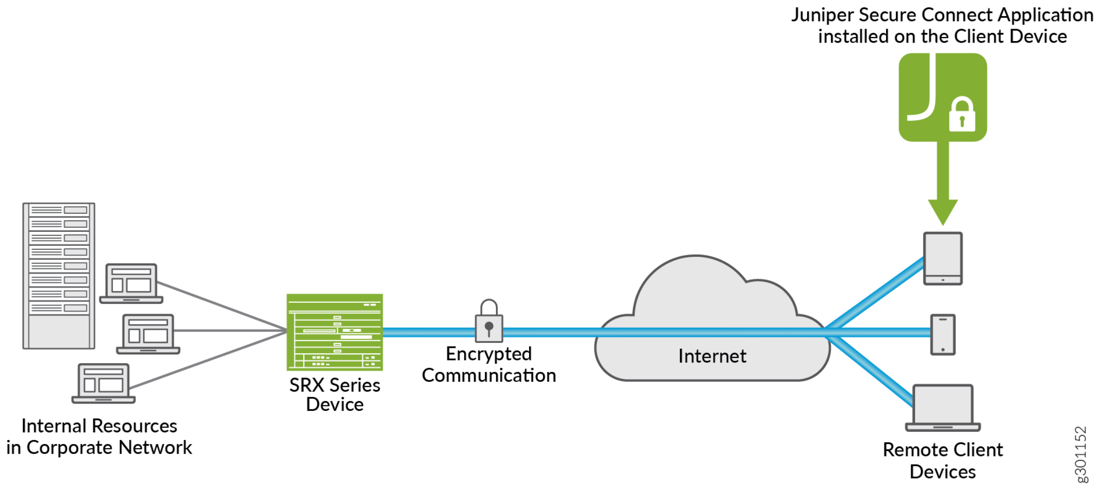
Juniper Secure Connect is a client-based SSL-VPN application that allows you to securely connect and access protected resources on your network. This application when combined with SRX Series Firewalls helps organizations quickly achieve dynamic, flexible, and adaptable connectivity from devices anywhere across the globe. Juniper Secure Connect extends visibility and enforcement from client to cloud using secure VPN connections.

Juniper Secure Connect application includes:

- SRX Series firewall—Serves as an entry and exit point for communication between users with Juniper Secure Connect and the protected resources on the corporate network or in cloud.
- Juniper Secure Connect application—Secures connectivity between the host clients running Microsoft Windows, Apple macOS, Google Android, and iOS operating systems and the protected resources. Juniper Secure Connect application connects through a VPN tunnel to the SRX Series Firewall to gain access to the protected resources in the network.

Figure 74 on page 1302 illustrates the Juniper Secure Connect remote access solution for establishing secure VPN connectivity for remote users at different locations.

Figure 74: Juniper Secure Connect Remote Access Solution


Juniper Secure Connect Application installed on the Client Device
Internal Resources in Corporate Network
SRX Series Device
Encrypted Communication
Internet
Remote Client Devices
g301152

To work with Juniper Secure Connect, you need SRX Series Firewall or vSRX Virtual Firewall instance running Junos OS Release 20.3R1 or later. See System Requirements.

Table 109: Features Support for Juniper Secure Connect

Feature	Description
Multi-Platform support	Supports Windows, macOS, Android, and iOS platforms.
Windows Pre-domain logon	Allows users to logon to the local Windows system through an already established VPN tunnel (using Windows Pre-Logon), so that it is authenticated to the central Windows domain or Active Directory.
Configuration support	Validates automatically that the most current policy is available before establishing the connection.
Biometric user authentication	Allows the user to protect their credentials using the operating system's built-in biometric authentication support.
Multi-Factor Authentication (MFA)	Allows you to use multi-factor authentication to extend the authentication.
Juniper Secure Connect license	Licenses are available in 1 year and 3 year subscription models.

Benefits of Juniper Secure Connect

- Secure remote access from anywhere with VPN
- Simple user experience
- Easy management of remote clients, policies, and VPN events from a single console (using J-Web)

WHAT'S NEXT

We recommend you to use J-Web wizard for Juniper Secure Connect configuration. For details on configuring Juniper Secure Connect, see Juniper Secure Connect Administrator Guide.

See Juniper Secure Connect User Guide for setting up client devices with Juniper Secure Connect application.

See these CLI configuration statements related to Juniper Secure Connect at: "default-profile" | **1481**, "windows-logon" | **1671**, "certificate" | **1458**, traceoptions, "profile" | **1589**, "global-options" | **1513**, "client-config" | **1463**, and "remote-access" | **1610**.

RELATED DOCUMENTATION

Overview

Migrating from Junos OS Dynamic VPN to Juniper Secure Connect

Preparing Juniper Secure Connect Configuration

14

CHAPTER

Quantum Safe VPN

Quantum Safe IPsec VPN

SUMMARY

Learn how to use and configure the out-of-band key retrieval mechanisms in the IKED process to negotiate with quantum secured IKE and IPsec SAs.

IN THIS SECTION

Quantum Safe IPsec VPN Overview

IN THIS SECTION

Quantum Security

The IPsec communication channel relies on the Internet Key Exchange (IKE) protocol. The IKE maintains security parameters to protect the data traffic. The security parameters include encryption and authentication algorithms, and associated keys.

The security protocols rely on asymmetric cryptographic algorithms such as Diffie Hellman (DH) or Elliptic Curve Diffie Hellman (ECDH) to establish keys are vulnerable to attacks.

To avoid security attacks, the RFC8784 introduces a method out-of-band method. The out-of-band method adds a secret key at the initiator and the responder. The secret key is Post-quantum Pre-shared Key (PPK).

- You can use the PPK in addition to the authentication method in IKEv2.

- PPK provides quantum resistance to any child SAs in initial negotiated IPsec SAs and any subsequent reeked IPsec SAs.
- With PPK and peer authentication key, initiator and responder can detect key mismatch.

Junos Key Manager Overview

You can use Junos Key Manager (JKM) to configure the static keys or dynamics keys to protect the data plane and control plane.

The JKM process acts as a key store and a proxy between the client or crypto application. The client or crypto application requires a key to establish an encrypted and authenticated quantum safe session with peer or application. The quantum safe uses the out-of-band key retrieval mechanism that lets two peers have the key. Different out-of-band mechanisms will have different protocols or methods to communicate. The JKM provides a common uniform interface for client or crypto applications to communicate.

Key Retrieval Mechanism

Two out-of-band key retrieval mechanisms in the IKED process to negotiate with quantum secured IKE and IPsec SAs.

- **Static Key**—With static key profiles, you can configure a static key ID and a corresponding key. The same static key ID and key gets generated every time a request to JKM over a static key profile.
- **Quantum Key Manager**—With quantum key manager key profiles, you can access the Quantum Key Distribution (QKD) devices and Quantum Network. The Quantum Network generates and exchange quantum keys between peers. Generates a different key ID and key every time on request to JKM over a quantum key manager key profile.

Use Key Profile for Quantum Safe IPsec VPN

With static key profiles, you can configure a static key ID and a corresponding key. To establish the quantum safe IPsec SAs, use the static key profile as Post-Quantum Pre-Shared Key (PPK) profile in the IPsec-VPN configuration. Uses the same key and key ID to re-authenticate existing IKE SA.

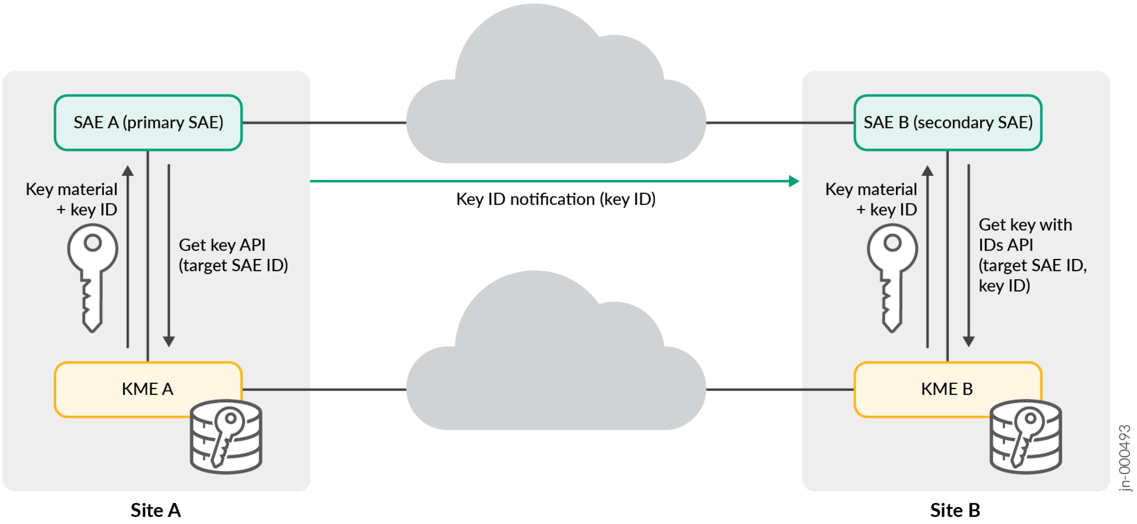
With quantum key manager key profile profiles, to access the Quantum Networks you need access to the QKD devices. The Quantum Network generates and exchanges quantum keys between peers. You can configure all the necessary parameters such as local SAE ID, URL to the QKD device, and so on. To establish IPsec SAs, use the quantum key manager key profile as Post-Quantum Pre-Shared Key (PPK) profile in the IPsec VPN configuration. Uses a different key and key ID to re-authenticate existing IKE SA.

Quantum Key Distribution

Quantum key distribution (QKD) is a secure key distribution method that uses quantum. Networks use quantum channels for generating the same key at both ends and monitor the quantum channel between the peers. These keys are dynamic, protects the data plane, and control plane.

Key Management Entity (KME) is the term we use to refer to the QKD devices on the management or control layer. QKD devices connect to each other through their quantum or QKD network. The KMEs connects over the public network through the secure channels for exchanging any control messages. The applications, Secure Application Entity (SAEs), and devices interact with KMEs through the secure channels as per ETSI specification. HTTPS combines with mutual TLS authentication and enables secure operations over the QKD network.

Figure 75: Two Devices Interacting with Their Corresponding QKD Devices to Establish a Quantum Secured Session



In the Figure 75 on page 1307, describes how the two devices interacting with their corresponding QKD devices to establish a quantum secured session

- SAE A role is primary. SAE A acts as the initiator to establish a quantum secured session with SAE B.
- The SAE B role is secondary. SAE B acts as the responder.
- The SAE A request the KME A through the Get key API to generate and share a new quantum key with SAE B with target SAE ID.

- The KME A performs the operation and responds to SAE A with the generated key ID and key material.
- KME B receives the key material and the generated ID key over the QKD network.
- The SAE A initiates secured session with SAE B directly using the same key and key ID.
- An exchange of messages establishes a secure session with SAE B.
- SAE A sends the key ID in plaintext or encrypted for the corresponding quantum key that is used to secure the session with SAE B.
- Once SAE B receives the key ID, the SAE B contacts KME B through the Get key with IDs API to get the corresponding quantum-key for the given key ID and target SAE ID or SAE A.
- After SAE B gets the key, a fully quantum secured session establishes between SAE A and SAE B.

Example: Configure Quantum-Secured IPsec AutoVPN Topology Using Quantum Key Manager Key Profile

SUMMARY

Learn how to configure IPsec AutoVPN topology using quantum key manager key profile.

IN THIS SECTION

Use this example to configure quantum-secured IPsec AutoVPN using quantum key manager key profile. The quantum key manager key profile includes parameters that are required to communicate with a Key Management Entity (KME) or Quantum Key Distribution (QKD) device. These parameters are as per ETSI GS QKD 014 specification.

In this example, we use the following devices:

- Hub as an IPsec VPN aggregator.
- Spoke 1 and Spoke 2 as remote sites.

The Hub, Spoke 1, and Spoke 2 use quantum key manager key profiles to communicate with KME Hub, KME Spoke 1, and KME Spoke 2 to fetch the QKD keys and establish then IPsec VPN tunnels.

TIP:

Table 110: Estimated Timers

Reading Time	Less than an hour.
Configuration Time	Less than an hour.

Example Prerequisites

Table 111: Requirements

Hardware requirements	Juniper Networks® SRX1500 Firewall or higher-numbered device models or Juniper Networks® vSRX Virtual Firewall (vSRX3.0).
Software requirements	Junos OS Release 22.4R1 or later with **JUNOS ike** and **JUNOS Key Manager** packages.

Before You Begin

Table 112: Let's Get Started

Benefits	With quantum-secured IPsec AutoVPN , you can: • Establish quantum-secured IKE or IPsec security associations (SAs) between a Hub and one or more Spokes with the help of a QKD device. • Extend the already standardized RFC 8784 procedure. • Use any QKD device supporting ETSI QKD Rest API.

Know more	IPsec VPN AutoVPN on Hub-and-Spoke Devices
Learn more	RFC 8784 - Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security ETSI QKD Rest API

Functional Overview

This section provides summary of the configuration components in this example.

Table 113: Detailed Configuration and Verification Procedures

Technologies used	To establish the quantum-safe IPsec tunnel, you must configure the following: • Key profile—Configure the following quantum key manager key profiles on the Hub. • *HUB_KM_PROFILE_1* • *SPOKE_1_KM_PROFILE_1* • *SPOKE_2_KM_PROFILE_1* Configure *SPOKE-1* and *SPOKE-2* for applications and services to retrieve QKD keys from external server. • IKE proposal—Configure the following IKE proposals on the Hub. • *HUB_IKE_PROP* • *SPOKE_1_IKE_PROP* • *SPOKE_2_IKE_PROP* Configure *SPOKE-1* and *SPOKE-2* with the required algorithms to establish an IKE SAs. • IKE policy—Configure the following IKE policies on the Hub. • *HUB_IKE_POL* • *SPOKE_1_IKE_POL* • *SPOKE_3_IKE_POL* Configure *SPOKE-1* and *SPOKE-2* to set the runtime negotiation and authentication attributes. • IKE gateway—Configure the following IKE gateways on the Hub. • *HUB_IKE_GW* • *SPOKE_1_IKE_GW* • *SPOKE_2_IKE_GW*

	Configure *SPOKE-1* and *SPOKE-2* to set the endpoints between the IPsec tunnels. A *ppk-profile* indicates which key-profile to use to establish quantum-safe IKE or IPsec SA. • IPsec proposal—Configure the following IPsec proposals on the Hub. • *HUB_IPSEC_PROP* • *SPOKE_1_IPSEC_PROP* • *SPOKE_2_IPSEC_PROP* Configure *SPOKE-1* and *SPOKE-2* with the required algorithms to establish an IPsec SA. • IPsec policy—Configure the following IPsec policies on the Hub. • *HUB_IPSEC_POL* • *SPOKE_1_IPSEC_POL* • *SPOKE_2_IPSEC_POL* Configure *SPOKE-1* and *SPOKE-2* to set the runtime IPsec negotiation attributes. • IPsec VPN—Configure the following IPsec VPNs on the Hub. • *HUB_IPSEC_VPN* • *SPOKE_1_IPSEC_VPN* • *SPOKE_2_IPSEC_VPN* Configure *SPOKE-1* and *SPOKE-2* to set the range of subnets that need to be secured. • Security zone—Configure three different security zones to segregate the traffic. • *trust* • *untrust*

	• *vpn* • Security policy—Configure the security policies *trust to vpn* and *vpn to trust* to select the type of data traffic that is secured through the IPsec SAs.
Primary verification tasks	Verify the IKE and IPsec SAs. Verify the established IKE and IPsec SAs are Quantum safe. Verify IPsec encryption and decryption statistics. Verify key profile statistics. Send data traffic from the host devices.

Table 114: Spoke 1: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/2.0	172.18.10.1/24
Interfaces	ge-0/0/1.0	192.168.80.1/24
Interfaces	st0.1 (tunnel interface)	family inet
Security zones	trust	The ge-0/0/1.0 interface is bound to this zone.
Security zones	untrust	The ge-0/0/2.0 interface is bound to this zone.
Security zones	vpn	The st0.1 interface is bound to this zone.

Table 114: Spoke 1: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Security policy	from-zone trust to-zone vpn	Match criteria: • Source-address: any • Destination-address: any • Application: any Action: permit
Security policy	from-zone vpn to-zone trust	Match criteria: • Source-address: any • Destination-address: any • Application: any Action: permit
CA profile	Root-CA	CA-identity: Root-CA URL: https://ca-server.juniper.net/certsrv/mscep/mscep.dll Revocation-check: disable
Key profile	SPOKE_1_KM_PROFILE_1	Key profile type: Quantum key manager URL: https://www.kme_spoke_1-qkd-server.net Local-sae-id: SAE_SPOKE_1 Local-certificate-id: SAE_SPOKE_1_CERT Trusted-cas: Root-CA

Table 114: Spoke 1: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
IKE Proposal	SPOKE_1_IKE_PROP	Authentication method: rsa-signatures DH group: group14 Authentication algorithm: sha-256 Encryption algorithm: aes-256-cbc Lifetime: 3600 seconds
IKE Policy	SPOKE_1_IKE_POL	Proposal reference: SPOKE_1_IKE_PROP Certificate reference: local-certificate SPOKE_1_CRT
IKE Gateway	SPOKE_1_IKE_GW	IKE policy reference: SPOKE_1_IKE_POL External interface: ge-0/0/2.0 Remote gateway address: 172.18.10.1 Local gateway address: 172.18.10.2 Version: v2-only ppk-profile: SPOKE_1_KM_PROFILE_1 Local-identity: distinguished-name Remote-identity: distinguished-name
IPsec Proposal	SPOKE_1_IPSEC_PROP	Protocol:esp Authentication-algorithm: hmac-sha-256-128 Encryption-algorithm: aes-256-cbc

Table 114: Spoke 1: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
IPsec Policy	IPSEC-POL	Proposal reference: SPOKE_1_IPSEC_PROP
IPsec VPN	VPN-to-HOST-2	IKE gateway reference: SPOKE_1_IKE_GW IPsec policy reference: SPOKE_1_IPSEC_POL Bind to interface: st0.1 Traffic-selector: ts1 and local-ip 192.168.80.0/24 remote-ip 192.168.90.0/24

Table 115: Spoke 2: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/2.0	172.18.10.3/24
Interfaces	ge-0/0/1.0	192.168.70.1/24
Interfaces	st0.1 (tunnel interface)	family inet
Security zones	trust	The ge-0/0/1.0 interface is bound to this zone.
Security zones	untrust	The ge-0/0/2.0 interface is bound to this zone.
Security zones	vpn	The st0.2 interface is bound to this zone.

Table 115: Spoke 2: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Security policy	from-zone trust to-zone vpn	Match criteria: • Source-address: any • Destination-address: any • Application: any Action: permit
Security policy	from-zone vpn to-zone trust	Match criteria: • Source-address: any • Destination-address: any • Application: any Action: permit
CA profile	Root-CA	CA-identity: Root-CA URL: https://ca-server.juniper.net/certsrv/mscep/mscep.dll Revocation-check: disable
Key profile	SPOKE_2_KM_PROFILE_1	Key profile type: Quantum key manager URL: https://www.kme_spoke_1-qkd-server.net Local-sae-id: SAE_SPOKE_2 Local-certificate-id: SAE_SPOKE_2_CERT Trusted-cas: Root-CA

Table 115: Spoke 2: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
IKE Proposal	SPOKE_2_IKE_PROP	Authentication method: rsa-signatures DH group: group14 Authentication algorithm: sha-256 Encryption algorithm: aes-256-cbc Lifetime: 3600 seconds
IKE Policy	SPOKE_2_IKE_POL	Proposal reference: SPOKE_2_IKE_PROP Certificate reference: local-certificate SPOKE_2_CRT
IKE Gateway	SPOKE_2_IKE_GW	IKE policy reference: SPOKE_2_IKE_POL External interface: ge-0/0/2.0 Remote gateway address: 172.18.10.1 Local gateway address: 172.18.10.3 Version: v2-only ppk-profile: SPOKE_2_KM_PROFILE_1 Local-identity: distinguished-name Remote-identity: distinguished-name
IPsec Proposal	SPOKE_2_IPSEC_PROP	Protocol:esp Authentication-algorithm: hmac-sha-256-128 Encryption-algorithm: aes-256-cbc

Table 115: Spoke 2: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
IPsec Policy	SPOKE_2_IPSEC_POL	Proposal reference: SPOKE_2_IPSEC_PROP
IPsec VPN	SPOKE_2_IPSEC_VPN	IKE gateway reference: SPOKE_2_IKE_GW IPsec policy reference: SPOKE_2_IPSEC_POL Bind to interface: st0.2 Traffic-selector: ts1 and local-ip 192.168.70.0/24 remote-ip 192.168.90.0/24

Table 116: Hub: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/2.0	172.18.10.1/24
Interfaces	ge-0/0/1.0	192.168.90.1/24
Interfaces	st0.1 (tunnel interface)	family inet
Security zones	trust	The ge-0/0/1.0 interface is bound to this zone.
Security zones	untrust	The ge-0/0/2.0 interface is bound to this zone.
Security zones	vpn	The st0.1 interface is bound to this zone.

Table 116: Hub: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Security policy	from-zone trust to-zone vpn	Match criteria: • Source-address: any • Destination-address: any • Application: any Action: permit
Security policy	From-zone vpn to-zone trust	Match criteria: • Source-address: any • Destination-address: any • Application: any Action: permit
CA profile	Root-CA	CA-identity: Root-CA URL: https://ca-server.juniper.net/certsrv/mscep/mscep.dll Revocation-check: disable
Key profile	HUB_KM_PROFILE_1	Key profile type: Quantum key manager URL: https://www.kme_spoke_1-qkd-server.net Local-sae-id: SAE_HUB Local-certificate-id: SAE_HUB_CERT Trusted-cas: Root-CA

Table 116: Hub: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
IKE Proposal	HUB_IKE_PROP	Authentication method: rsa-signatures DH group: group14 Authentication algorithm: sha-256 Encryption algorithm: aes-256-cbc Lifetime: 3600 seconds
IKE Policy	HUB_IKE_POL	Proposal reference: HUB_IKE_PROP Certificate reference: local-certificate HUB_CRT
IKE Gateway	HUB_IKE_GW	IKE policy reference: HUB_IKE_POL External interface: ge-0/0/2.0 Local gateway address: 172.18.10.1 Version: v2-only ppk-profile: HUB_KM_PROFILE_1 Local-identity: distinguished-name Dynamic gateway: • Remote-identity: distinguished-name wildcard "C=us DC=juniper" • ike-user-type: group-ike-id

Table 116: Hub: Interface, Security Zone, Security Policy, Key Profile, PKI, IKE and IPsec Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
IPsec Proposal	HUB_IPSEC__PROP	Protocol:esp Authentication-algorithm: hmac-sha-256-128 Encryption-algorithm: aes-256-cbc
IPsec Policy	HUB_IPSEC_POL	Proposal reference: HUB_IPSEC_PROP
IPsec VPN	HUB_IPSEC_VPN	IKE gateway reference: HUB_IKE_GW IPsec policy reference: HUB_IPSEC_POL Bind to interface: st0.1 Traffic-selector: ts1 and local-ip 192.168.90.0/24 remote-ip 0.0.0.0/0

Table 117: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
The security policy permits traffic from the trust zone to the VPN zone.	VPN-OUT	Match criteria: • source-address HOST-1-Net • destination-address HOST-2-Net • application any Action: permit

Table 117: Security Policy Configuration Parameters *(Continued)*

Purpose	Name	Configuration Parameters
The security policy permits traffic from the VPN zone to the trust zone.	VPN-IN	Match criteria: • source-address HOST-2-Net • destination-address HOST-1-Net • application any Action: permit

Topology Overview

In this example, Spoke 1 and Spoke 2 initiate the negotiation of Quantum-safe IPsec tunnels with the Hub using QKD keys from KME Spoke 1 and KME Spoke 2. The Hub responds to the requests by verifying Spoke 1 and Spoke 2 identities and keys from KME Hub to establish Quantum-safe IPsec VPN tunnels. After IPsec tunnels are established, the data traffic between Host 1 and Host 3, and that between Host 2 and Host 3 is Quantum secured.

Table 118: Devices, Role, and Functionalities Used in This Configuration

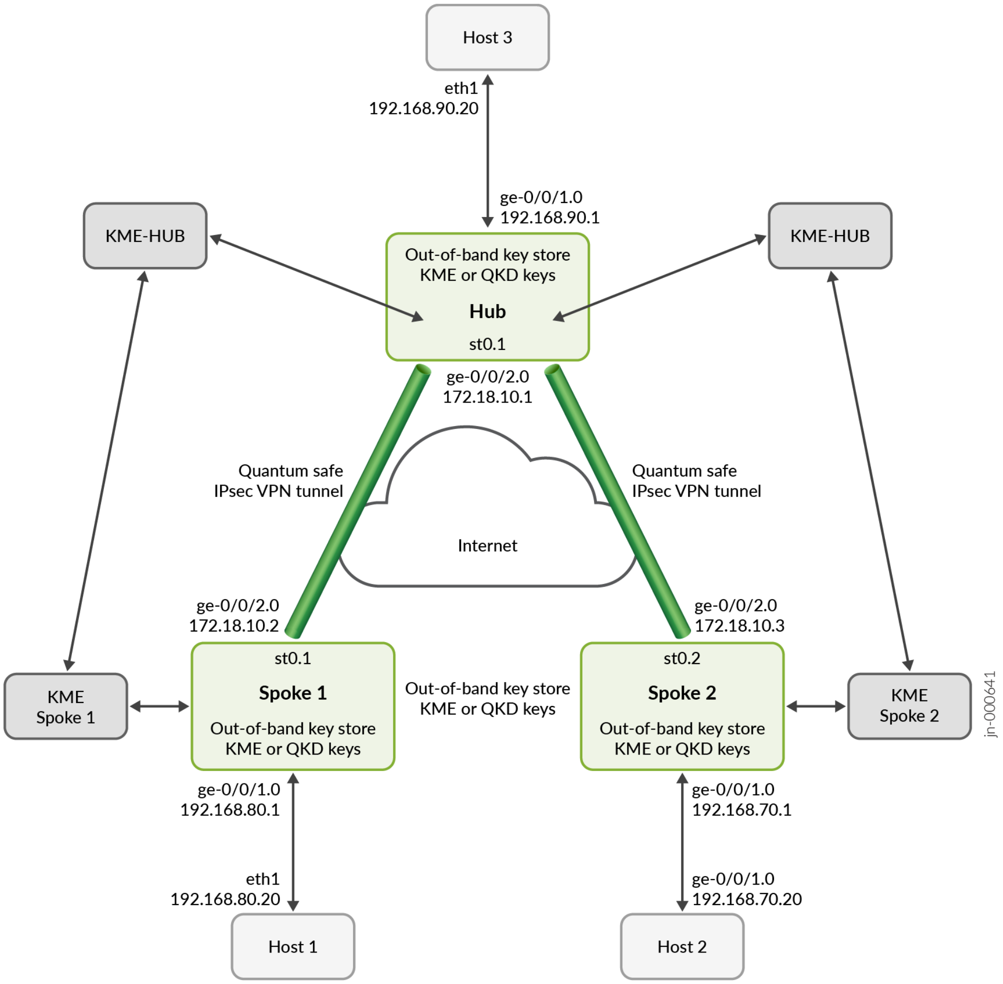
Hostname	Role	Function
`HUB`	SRX Series Firewall capable of establishing IPsec tunnels	Responds to IKE or IPsec SA negotiation and establishes Quantum-safe IPsec tunnels using QKD key from KME-HUB QKD device on `SPOKE-1` and `SPOKE-2`.
`SPOKE-1`	SRX Series Firewall capable of establishing IPsec tunnels	Initiates IKE or IPsec SA negotiation and establishes Quantum-safe IPsec tunnels with hub using QKD key from `KME-SPOKE-1` QKD device

Table 118: Devices, Role, and Functionalities Used in This Configuration *(Continued)*

Hostname	Role	Function
SPOKE-2	SRX Series Firewall capable of establishing IPsec tunnels	Initiates IKE or IPsec SA negotiation and establishes Quantum-safe IPsec tunnels with hub using QKD key from KME-SPOKE-2 QKD device
HOST-1	Host inside the trusted zone or LAN side of SPOKE 1	Initiates client-side traffic towards HOST-3
HOST-2	Host inside the trusted zone or LAN side of SPOKE 2	Initiates client-side traffic towards HOST-3
HOST- 3	Host inside the trusted zone or LAN side of hub	Responds to client-side traffic from HOST-1 and HOST-2
KME-HUB	Third-party QKD device	Provides QKD keys in response to key requests from HUB
KME-SPOKE-1	Third-party QKD device	Provides QKD keys in response to key requests from SPOKE-1
KME-SPOKE-2	Third-party QKD device	Provides QKD keys in response to key requests from SPOKE-2

Topology Illustration

Figure 76: Quantum Key Manager with AutoVPN



Step-By-Step Configuration on Device-Under-Test (DUT)

IN THIS SECTION

NOTE: For complete sample configurations on the DUT, see:

- "Appendix 1: Set Commands on all Devices" on page 1344

Configure Hub

1. Configure interfaces.

```
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.1/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.90.1/24
set interfaces st0 unit 1 family inet
```

2. Configure security zones.

```
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
```

3. Configure security policies.

```
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address
any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address
any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

4. Configure the CA profile and the CA certificate.

```
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/
```

```
mscep/mscep.dll
set security pki ca-profile Root-CA revocation-check disable
```

5. Bind the CA certificate to the CA profile.

```
request security pki ca-certificate enroll ca-profile Root-CA
```

6. Configure local certificates.

```


request security pki generate-key-pair certificate-id HUB_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id HUB_CRT challenge-password
<password> domain-name hub.juniper.net email hub@juniper.net subject
DC=juniper,CN=hub.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
request security pki local-certificate load certificate-id SAE_HUB filename SAE_HUB.cert
key SAE_HUB.key
```

7. Configure the quantum key manager key profile.

```
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager url https://
www.kme_hub-qkd-server.net
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager local-sae-id SAE_HUB
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager local-certificate-id
SAE_HUB_CERT
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager trusted-cas Root-CA
```

8. Configure the IKE proposal.

```
set security ike proposal HUB_IKE_PROP authentication-method rsa-signatures
set security ike proposal HUB_IKE_PROP dh-group group14
set security ike proposal HUB_IKE_PROP authentication-algorithm sha-256
set security ike proposal HUB_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal HUB_IKE_PROP lifetime-seconds 3600
```

9. Configure the IKE policy.

```
set security ike policy HUB_IKE_POL proposals HUB_IKE_PROP
set security ike policy HUB_IKE_POL certificate local-certificate HUB_CRT
```

10. Configure the IKE gateway.

```
set security ike gateway HUB_IKE_GW local-address 172.18.10.1
set security ike gateway HUB_IKE_GW ike-policy HUB_IKE_POL
set security ike gateway HUB_IKE_GW external-interface ge-0/0/2.0
set security ike gateway HUB_IKE_GW local-identity distinguished-name
set security ike gateway HUB_IKE_GW dynamic ike-user-type group-ike-id
set security ike gateway HUB_IKE_GW dynamic distinguished-name wildcard C=us,DC=juniper
set security ike gateway HUB_IKE_GW version v2-only
```

11. Bind the quantum key manager key profile as the IKE gateway ppk-profile.

```

set security ike gateway HUB_IKE_GW ppk-profile HUB_KM_PROFILE_1
```

12. Configure the IPsec proposal.

```
set security ipsec proposal HUB_IPSEC_PROP protocol esp
set security ipsec proposal HUB_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal HUB_IPSEC_PROP encryption-algorithm aes-256-cbc
```

13. Configure the IPsec policy.

```
set security ipsec policy HUB_IPSEC_POL proposals HUB_IPSEC_PROP
```

14. Configure the IPsec VPN.

```
set security ipsec vpn HUB_IPSEC_VPN bind-interface st0.1
set security ipsec vpn HUB_IPSEC_VPN ike gateway HUB_IKE_GW
set security ipsec vpn HUB_IPSEC_VPN ike ipsec-policy HUB_IPSEC_POL
set security ipsec vpn HUB_IPSEC_VPN traffic-selector ts1 local-ip 192.168.90.0/24
set security ipsec vpn HUB_IPSEC_VPN traffic-selector ts1 remote-ip 0.0.0.0/0
```

If you are done configuring the device, enter `commit` from configuration mode.

Configure Spoke 1

1. Configure interfaces.

   ```
   set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.2/24
   set interfaces ge-0/0/1 unit 0 family inet address 192.168.80.1/24
   set interfaces st0 unit 1 family inet
   ```

2. Configure security zones.

   ```
   set security zones security-zone untrust host-inbound-traffic system-services ike
   set security zones security-zone untrust interfaces ge-0/0/2.0
   set security zones security-zone vpn interfaces st0.1
   set security zones security-zone trust host-inbound-traffic system-services ping
   set security zones security-zone trust interfaces ge-0/0/1.0
   ```

3. Configure security policies.

   ```
   set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
   set security policies from-zone trust to-zone vpn policy vpn_out match destination-address
   any
   set security policies from-zone trust to-zone vpn policy vpn_out match application any
   set security policies from-zone trust to-zone vpn policy vpn_out then permit
   set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
   set security policies from-zone vpn to-zone trust policy vpn_in match destination-address
   any
   set security policies from-zone vpn to-zone trust policy vpn_in match application any
   set security policies from-zone vpn to-zone trust policy vpn_in then permit
   ```

4. Configure the CA profile and the CA certificate.

   ```
   set security pki ca-profile Root-CA ca-identity Root-CA
   set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/
   ```

```
mscep/mscep.dll
set security pki ca-profile Root-CA revocation-check disable
```

5. Bind the CA certificate to the CA profile.

```
request security pki ca-certificate enroll ca-profile Root-CA
```

6. Configure local certificates.

```
request security pki generate-key-pair certificate-id SPOKE_1_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id SPOKE_1_CRT challenge-password
<password> domain-name spoke_1.juniper.net email spoke_1@juniper.net subject
DC=juniper,CN=spoke_1.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
request security pki local-certificate load certificate-id SAE_SPOKE_1 filename
SAE_SPOKE_1.cert key SAE_SPOKE_1.key
```

7. Configure the quantum key manager key profile.

```
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager url https://
www.kme_spoke_1-qkd-server.net
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager local-sae-id
SAE_SPOKE_1
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager local-
certificate-id SAE_SPOKE_1_CERT
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager trusted-cas Root-
CA
```

8. Configure the IKE proposal.

```
set security ike proposal SPOKE_1_IKE_PROP authentication-method rsa-signatures
set security ike proposal SPOKE_1_IKE_PROP dh-group group14
set security ike proposal SPOKE_1_IKE_PROP authentication-algorithm sha-256
set security ike proposal SPOKE_1_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal SPOKE_1_IKE_PROP lifetime-seconds 3600
```

9. Configure the IKE policy.

```
set security ike policy SPOKE_1_IKE_POL proposals SPOKE_1_IKE_PROP
set security ike policy SPOKE_1_IKE_POL certificate local-certificate SPOKE_1_CRT
```

10. Configure the IKE gateway.

```
set security ike gateway SPOKE_1_IKE_GW address 172.18.10.1
set security ike gateway SPOKE_1_IKE_GW local-address 172.18.10.2
set security ike gateway SPOKE_1_IKE_GW ike-policy SPOKE_1_IKE_POL
set security ike gateway SPOKE_1_IKE_GW external-interface ge-0/0/2.0
set security ike gateway SPOKE_1_IKE_GW local-identity distinguished-name
set security ike gateway SPOKE_1_IKE_GW remote-identity distinguished-name
set security ike gateway SPOKE_1_IKE_GW version v2-only
```

11. Bind the quantum key manager key profile as the IKE gateway ppk-profile.

```
set security ike gateway SPOKE_1_IKE_GW ppk-profile SPOKE_1_KM_PROFILE_1
```

12. Configure the IPsec proposal.

```
set security ipsec proposal SPOKE_1_IPSEC_PROP protocol esp
set security ipsec proposal SPOKE_1_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal SPOKE_1_IPSEC_PROP encryption-algorithm aes-256-cbc
```

13. Configure the IPsec policy.

```
set security ipsec policy SPOKE_1_IPSEC_POL proposals SPOKE_1_IPSEC_PROP
```

14. Configure the IPsec VPN.

```
set security ipsec vpn SPOKE_1_IPSEC_VPN bind-interface st0.1
set security ipsec vpn SPOKE_1_IPSEC_VPN ike gateway SPOKE_1_IKE_GW
set security ipsec vpn SPOKE_1_IPSEC_VPN ike ipsec-policy SPOKE_1_IPSEC_POL
set security ipsec vpn SPOKE_1_IPSEC_VPN traffic-selector ts1 local-ip 192.168.80.0/24
set security ipsec vpn SPOKE_1_IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
```

If you are done configuring the device, enter `commit` from configuration mode.

Configure Spoke 2

1. Configure interfaces.

```
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.3/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.70.1/24
set interfaces st0 unit 2 family inet
```

2. Configure security zones.

```
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.2
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
```

3. Configure security policies.

```
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address
any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address
any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

4. Configure the CA profile and the CA certificate.

```
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/
mscep/mscep.dll
set security pki ca-profile Root-CA revocation-check disable
```

5. Bind the CA certificate to the CA profile.

```
request security pki ca-certificate enroll ca-profile Root-CA
```

6. Configure the local certificates.

```
request security pki generate-key-pair certificate-id SPOKE_2_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id SPOKE_2_CRT challenge-password
<password> domain-name spoke_2.juniper.net email spoke_2@juniper.net subject
DC=juniper,CN=spoke_2.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
request security pki local-certificate load certificate-id SAE_SPOKE_2 filename
SAE_SPOKE_2.cert key SAE_SPOKE_2.key
```

7. Configure the quantum key manager key profile.

```
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager url https://
www.kme_spoke_2-qkd-server.net
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager local-sae-id
SAE_SPOKE_2
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager local-
certificate-id SAE_SPOKE_2_CERT
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager trusted-cas Root-
CA
```

8. Configure the IKE proposal.

```
set security ike proposal SPOKE_2_IKE_PROP authentication-method rsa-signatures
set security ike proposal SPOKE_2_IKE_PROP dh-group group14
set security ike proposal SPOKE_2_IKE_PROP authentication-algorithm sha-256
set security ike proposal SPOKE_2_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal SPOKE_2_IKE_PROP lifetime-seconds 3600
```

9. Configure the IKE policy.

```
set security ike policy SPOKE_2_IKE_POL proposals SPOKE_2_IKE_PROP
set security ike policy SPOKE_2_IKE_POL certificate local-certificate SPOKE_2_CRT
```

10. Configure the IKE gateway.

```
set security ike gateway SPOKE_2_IKE_GW address 172.18.10.1
set security ike gateway SPOKE_2_IKE_GW local-address 172.18.10.3
set security ike gateway SPOKE_2_IKE_GW ike-policy SPOKE_2_IKE_POL
set security ike gateway SPOKE_2_IKE_GW external-interface ge-0/0/2.0
set security ike gateway SPOKE_2_IKE_GW local-identity distinguished-name
set security ike gateway SPOKE_2_IKE_GW remote-identity distinguished-name
set security ike gateway SPOKE_2_IKE_GW version v2-only
```

11. Bind the quantum key manager key profile as the IKE gateway ppk-profile.

```
set security ike gateway SPOKE_2_IKE_GW ppk-profile SPOKE_2_KM_PROFILE_1
```

12. Configure the IPsec proposal.

```
set security ipsec proposal SPOKE_2_IPSEC_PROP protocol esp
set security ipsec proposal SPOKE_2_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal SPOKE_2_IPSEC_PROP encryption-algorithm aes-256-cbc
```

13. Configure the IPsec policy.

```
set security ipsec policy SPOKE_2_IPSEC_POL proposals SPOKE_2_IPSEC_PROP
```

14. Configure the IPsec VPN.

```
set security ipsec vpn SPOKE_2_IPSEC_VPN bind-interface st0.2
set security ipsec vpn SPOKE_2_IPSEC_VPN ike gateway SPOKE_2_IKE_GW
set security ipsec vpn SPOKE_2_IPSEC_VPN ike ipsec-policy SPOKE_2_IPSEC_POL
set security ipsec vpn SPOKE_2_IPSEC_VPN traffic-selector ts1 local-ip 192.168.70.0/24
set security ipsec vpn SPOKE_2_IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

This section provides a list of show commands that you can use to verify the feature in this example.

Table 119: Verification Tasks

Command	Verification Task
`ping 192.168.90.20 source 192.168.80.20 count 4`	Ping from Host-1 to Host 3.
`ping 192.168.90.20 source 192.168.70.20 count 4`	Ping from Host 2 to Host 3.
`show security ike security-associations detail`	Verify the IKE SAs.
`show security ipsec security-associations detail`	Verify the IPsec SAs.
`show security ipsec statistics`	Verify IPsec encryption and decryption statistics.
`show security key-manager profiles detail`	Verify key profile statistics.

Ping from Host 1 to Host 3

Purpose

Verify the connectivity from Host 1 to Host 3.

Action

From operational mode, enter the `ping 192.168.90.20 source 192.168.80.20 count 5` to view the connectivity from Host 1 to Host 3.

```
user@host1# ping 192.168.90.20 source 192.168.80.20 count 5
PING 192.168.90.20 (192.168.90.20): 56 data bytes
64 bytes from 192.168.90.20: icmp_seq=0 ttl=64 time=2.151 ms
64 bytes from 192.168.90.20: icmp_seq=1 ttl=64 time=1.710 ms
64 bytes from 192.168.90.20: icmp_seq=2 ttl=64 time=1.349 ms
64 bytes from 192.168.90.20: icmp_seq=3 ttl=64 time=1.597 ms
64 bytes from 192.168.90.20: icmp_seq=4 ttl=64 time=1.515 ms
--- 192.168.90.20 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.349/1.702/2.151/0.290 ms

Data traffic is successfully flowing between the HOSTs
```

Meaning

The sample output confirms the connectivity from Host 1 to Host 3.

Ping from Host 2 to Host 3

Purpose

Verify the connectivity from Host 2 to Host 3.

Action

From operational mode, enter the `ping 192.168.90.20 source 192.168.80.20 count 5` command to view the connectivity from Host 2 to Host 3.

```
user@host2# ping 192.168.90.20 source 192.168.70.20 count 5
PING 192.168.90.20 (192.168.90.20): 56 data bytes
64 bytes from 192.168.90.20: icmp_seq=0 ttl=64 time=2.151 ms
64 bytes from 192.168.90.20: icmp_seq=1 ttl=64 time=1.710 ms
64 bytes from 192.168.90.20: icmp_seq=2 ttl=64 time=1.349 ms
64 bytes from 192.168.90.20: icmp_seq=3 ttl=64 time=1.597 ms
64 bytes from 192.168.90.20: icmp_seq=4 ttl=64 time=1.759 ms
```

```
--- 192.168.90.20 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.349/1.702/2.151/0.290 ms

Data traffic is successfully flowing between the HOSTs
```

Meaning

The sample output confirms the connectivity from Host 2 to Host 3.

Verify IKE SAs

Purpose

Verify the IKE SAs.

Action

From operational mode, enter the `show security ike security-associations detail` command to view the IKE SAs.

```
user@hub> show security ike security-associations detail

IKE peer 172.18.10.3, Index 2161, Gateway Name: HUB_IKE_GW
  Role: Responder, State: UP
  Initiator cookie: bccc74c70f0b81b9, Responder cookie: 872d364f15b29c28
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local gateway interface: ge-0/0/2.0
  Routing instance: default
  Local: 172.18.10.1:500, Remote: 172.18.10.3:500
  Lifetime: Expires in 3464 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security,
CN=spoke_2.juniper.net
  AAA assigned IP: 0.0.0.0
  PPK-profile: HUB_KM_PROFILE_1
     Optional: No
     State   : Used
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :                 2661
   Output bytes  :                 2586
   Input  packets:                    5
   Output packets:                    5
   Input  fragmented packets:       4
   Output fragmented packets:       4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500446

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 172.18.10.1:500, Remote: 172.18.10.3:500
    Local identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security, 
CN=hub.juniper.net
    Remote identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security, 
CN=spoke_2.juniper.net
    Flags: IKE SA is created

  IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                  Responder stats:
    Request Out             : 0                       Request In             : 
0
    Response In             : 0                       Response Out           : 
0
    No Proposal Chosen In   : 0                       No Proposal Chosen Out : 
0
    Invalid KE In           : 0                       Invalid KE Out         : 
0
    TS Unacceptable In      : 0                       TS Unacceptable Out    : 
0
    Res DH Compute Key Fail : 0                       Res DH Compute Key Fail: 
0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0

IKE peer 172.18.10.2, Index 2162, Gateway Name: HUB_IKE_GW
  Role: Responder, State: UP
  Initiator cookie: 5e17d5924c619788, Responder cookie: 15f1e3c4252ba6f8
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local gateway interface: ge-0/0/2.0
  Routing instance: default
  Local: 172.18.10.1:500, Remote: 172.18.10.2:500
  Lifetime: Expires in 3464 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security, 
CN=spoke.juniper.net
  AAA assigned IP: 0.0.0.0
  PPK-profile: HUB_KM_PROFILE_1
     Optional: No
     State   : Used
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :                 2645
   Output bytes  :                 2586
   Input  packets:                    5
   Output packets:                    5
   Input  fragmented packets:       4
   Output fragmented packets:       4
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500447

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 172.18.10.1:500, Remote: 172.18.10.2:500
    Local identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security, 
CN=hub.juniper.net
    Remote identity: C=us, DC=juniper, ST=california, L=sunnyvale, O=juniper, OU=security, 
CN=spoke.juniper.net
    Flags: IKE SA is created

  IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                  Responder stats:
    Request Out             : 0                       Request In              : 
0
```

```
    Response In             : 0                       Response Out           :
0
    No Proposal Chosen In   : 0                       No Proposal Chosen Out :
0
    Invalid KE In           : 0                       Invalid KE Out         :
0
    TS Unacceptable In      : 0                       TS Unacceptable Out    :
0
    Res DH Compute Key Fail : 0                       Res DH Compute Key Fail:
0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0
```

Meaning

The sample output confirms the IKE SAs.

Verify IPsec SAs

Purpose

Verify the IPsec SAs.

Action

From operational mode, enter the `show security ipsec security-associations detail` command to view the IPsec SAs.

```
user@hub> show security ipsec security-associations detail


ID: 500446 Virtual-system: root, VPN Name: HUB_IPSEC_VPN
  Local Gateway: 172.18.10.1, Remote Gateway: 172.18.10.3
  Traffic Selector Name: ts1
  Local Identity: ipv4(192.168.90.0-192.168.90.255)
  Remote Identity: ipv4(192.168.70.0-192.168.70.255)
  TS Type: traffic-selector
  Version: IKEv2
  Quantum Secured: Yes
  PFS group: N/A
  Passive mode tunneling: Disabled
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: HUB_IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Tunnel events:
    Fri Jul 21 2023 00:31:08: IPsec SA negotiation succeeds (1 times)
  Location: FPC 0, PIC 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0xcf48c0c9, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3464 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2778 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
    IKE SA Index: 2161
  Direction: outbound, SPI: 0x86c9ba76, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3464 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2778 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
    IKE SA Index: 2161

ID: 500447 Virtual-system: root, VPN Name: HUB_IPSEC_VPN
  Local Gateway: 172.18.10.1, Remote Gateway: 172.18.10.2
  Traffic Selector Name: ts1
  Local Identity: ipv4(192.168.90.0-192.168.90.255)
  Remote Identity: ipv4(192.168.80.0-192.168.80.255)
  TS Type: traffic-selector
  Version: IKEv2
  Quantum Secured: Yes
  PFS group: N/A
  Passive mode tunneling: Disabled
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: HUB_IPSEC_POL
```

```
Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
Tunnel events:
  Fri Jul 21 2023 00:31:08: IPsec SA negotiation succeeds (1 times)
Location: FPC 0, PIC 0
Anchorship: Thread 1
Distribution-Profile: default-profile
Direction: inbound, SPI: 0x4275d756, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3464 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 2772 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64
  Extended-Sequence-Number: Disabled
  tunnel-establishment: establish-tunnels-responder-only
  IKE SA Index: 2162
Direction: outbound, SPI: 0xe37b5568, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3464 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 2772 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64
  Extended-Sequence-Number: Disabled
  tunnel-establishment: establish-tunnels-responder-only
  IKE SA Index: 2162
```

Meaning

The sample output confirms the IPsec SAs.

Verify IPsec Statistics

Purpose

Verify the IPsec statistics.

Action

From operational mode, enter the `show security ipsec statistics` command to view the IPsec statistics.

```
user@hub> show security ipsec statistics

ESP Statistics:
  Encrypted bytes:             1560
  Decrypted bytes:             1560
  Encrypted packets:             10
  Decrypted packets:             10
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
  Invalid SPI: 0, TS check fail: 0
  Exceeds tunnel MTU: 0
  Discarded: 0
```

Meaning

The sample output confirms the IPsec statistics.

Verify Key Manager Profile

Purpose

Verify the key manager profile.

Action

From operational mode, enter the `show security key-manager profiles detail` command to view the quantum key manager profile.

```
user@hub> show security key-manager profiles detail

Name: HUB_KM_PROFILE_1, Index: 6, Type: Quantum-key-manager
  Configured-at: 21.07.23 (00:14:00)
  Time-elapsed: 0 hrs 19 mins 24 secs
  Url: https://kme.juniper.net:8080
  Local-sae-id: SAE_HUB
  Local-certificate-id: SAE_HUB_CERT
  Trusted-cas: [ ROOT_CA_CERT ]
  Peer-sae-ids: N/A
  Default-key-size: N/A
  Request stats:
    Received: 2
    In-progress: 0
    Success: 2
    Failed: 0
```

Meaning

The sample output confirms the quantum key manager profile.

Appendix 1: Set Commands on all Devices

Set command output on all devices.

Set Commands on Hub

```
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
set security pki ca-profile Root-CA revocation-check disable
request security pki ca-certificate enroll ca-profile Root-CA
request security pki generate-key-pair certificate-id HUB_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id HUB_CRT challenge-password
<password> domain-name hub.juniper.net email hub@juniper.net subject
DC=juniper,CN=hub.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-profile
```

```
Root-CA
request security pki local-certificate load certificate-id SAE_HUB filename SAE_HUB.cert key
SAE_HUB.key
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager url https://www.kme_hub-
qkd-server.net
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager local-sae-id SAE_HUB
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager local-certificate-id
SAE_HUB_CERT
set security key-manager profiles HUB_KM_PROFILE_1 quantum-key-manager trusted-cas Root-CA
set security ike proposal HUB_IKE_PROP authentication-method rsa-signatures
set security ike proposal HUB_IKE_PROP dh-group group14
set security ike proposal HUB_IKE_PROP authentication-algorithm sha-256
set security ike proposal HUB_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal HUB_IKE_PROP lifetime-seconds 3600
set security ike policy HUB_IKE_POL proposals HUB_IKE_PROP
set security ike policy HUB_IKE_POL certificate local-certificate HUB_CRT
set security ike gateway HUB_IKE_GW local-address 172.18.10.1
set security ike gateway HUB_IKE_GW ike-policy HUB_IKE_POL
set security ike gateway HUB_IKE_GW external-interface ge-0/0/2.0
set security ike gateway HUB_IKE_GW local-identity distinguished-name
set security ike gateway HUB_IKE_GW dynamic ike-user-type group-ike-id
set security ike gateway HUB_IKE_GW dynamic distinguished-name wildcard C=us,DC=juniper
set security ike gateway HUB_IKE_GW ppk-profile HUB_KM_PROFILE_1
set security ike gateway HUB_IKE_GW version v2-only
set security ipsec proposal HUB_IPSEC_PROP protocol esp
set security ipsec proposal HUB_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal HUB_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy HUB_IPSEC_POL proposals HUB_IPSEC_PROP
set security ipsec vpn HUB_IPSEC_VPN bind-interface st0.1
set security ipsec vpn HUB_IPSEC_VPN ike gateway HUB_IKE_GW
set security ipsec vpn HUB_IPSEC_VPN ike ipsec-policy HUB_IPSEC_POL
set security ipsec vpn HUB_IPSEC_VPN traffic-selector ts1 local-ip 192.168.90.0/24
set security ipsec vpn HUB_IPSEC_VPN traffic-selector ts1 remote-ip 0.0.0.0/0
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.1/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.90.1/24
set interfaces st0 unit 1 family inet
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

Set Commands on Spoke 1

```
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
set security pki ca-profile Root-CA revocation-check disable
request security pki ca-certificate enroll ca-profile Root-CA
request security pki generate-key-pair certificate-id SPOKE_1_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id SPOKE_1_CRT challenge-password
<password> domain-name spoke_1.juniper.net email spoke_1@juniper.net subject
DC=juniper,CN=spoke_1.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
request security pki local-certificate load certificate-id SAE_SPOKE_1 filename SAE_SPOKE_1.cert
key SAE_SPOKE_1.key
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager url https://
www.kme_spoke_1-qkd-server.net
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager local-sae-id
SAE_SPOKE_1
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager local-certificate-id
SAE_SPOKE_1_CERT
set security key-manager profiles SPOKE_1_KM_PROFILE_1 quantum-key-manager trusted-cas Root-CA
set security ike proposal SPOKE_1_IKE_PROP authentication-method rsa-signatures
set security ike proposal SPOKE_1_IKE_PROP dh-group group14
set security ike proposal SPOKE_1_IKE_PROP authentication-algorithm sha-256
set security ike proposal SPOKE_1_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal SPOKE_1_IKE_PROP lifetime-seconds 3600
set security ike policy SPOKE_1_IKE_POL proposals SPOKE_1_IKE_PROP
set security ike policy SPOKE_1_IKE_POL certificate local-certificate SPOKE_1_CRT
set security ike gateway SPOKE_1_IKE_GW address 172.18.10.1
set security ike gateway SPOKE_1_IKE_GW local-address 172.18.10.2
set security ike gateway SPOKE_1_IKE_GW ike-policy SPOKE_1_IKE_POL
set security ike gateway SPOKE_1_IKE_GW external-interface ge-0/0/2.0
set security ike gateway SPOKE_1_IKE_GW local-identity distinguished-name
set security ike gateway SPOKE_1_IKE_GW remote-identity distinguished-name
set security ike gateway SPOKE_1_IKE_GW ppk-profile SPOKE_1_KM_PROFILE_1
set security ike gateway SPOKE_1_IKE_GW version v2-only
set security ipsec proposal SPOKE_1_IPSEC_PROP protocol esp
set security ipsec proposal SPOKE_1_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal SPOKE_1_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy SPOKE_1_IPSEC_POL proposals SPOKE_1_IPSEC_PROP
set security ipsec vpn SPOKE_1_IPSEC_VPN bind-interface st0.1
set security ipsec vpn SPOKE_1_IPSEC_VPN ike gateway SPOKE_1_IKE_GW
set security ipsec vpn SPOKE_1_IPSEC_VPN ike ipsec-policy SPOKE_1_IPSEC_POL
set security ipsec vpn SPOKE_1_IPSEC_VPN traffic-selector ts1 local-ip 192.168.80.0/24
set security ipsec vpn SPOKE_1_IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.2/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.80.1/24
set interfaces st0 unit 1 family inet
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.1
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

Set Commands on Spoke 2

```
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url https://ca-server.juniper.net/certsrv/mscep/
mscep.dll
set security pki ca-profile Root-CA revocation-check disable
request security pki ca-certificate enroll ca-profile Root-CA
request security pki generate-key-pair certificate-id SPOKE_2_CRT size 2048 type rsa
request security pki local-certificate enroll certificate-id SPOKE_2_CRT challenge-password
<password> domain-name spoke_2.juniper.net email spoke_2@juniper.net subject
DC=juniper,CN=spoke_2.juniper.net,OU=security,O=juniper,L=sunnyvale,ST=california,C=us ca-
profile Root-CA
request security pki local-certificate load certificate-id SAE_SPOKE_2 filename SAE_SPOKE_2.cert
```

```
key SAE_SPOKE_2.key
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager url https://
www.kme_spoke_2-qkd-server.net
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager local-sae-id
SAE_SPOKE_2
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager local-certificate-id
SAE_SPOKE_2_CERT
set security key-manager profiles SPOKE_2_KM_PROFILE_1 quantum-key-manager trusted-cas Root-CA
set security ike proposal SPOKE_2_IKE_PROP authentication-method rsa-signatures
set security ike proposal SPOKE_2_IKE_PROP dh-group group14
set security ike proposal SPOKE_2_IKE_PROP authentication-algorithm sha-256
set security ike proposal SPOKE_2_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal SPOKE_2_IKE_PROP lifetime-seconds 3600
set security ike policy SPOKE_2_IKE_POL proposals SPOKE_IKE_PROP
set security ike policy SPOKE_2_IKE_POL certificate local-certificate SPOKE_2_CRT
set security ike gateway SPOKE_2_IKE_GW address 172.18.10.1
set security ike gateway SPOKE_2_IKE_GW local-address 172.18.10.3
set security ike gateway SPOKE_2_IKE_GW ike-policy SPOKE_2_IKE_POL
set security ike gateway SPOKE_2_IKE_GW external-interface ge-0/0/2.0
set security ike gateway SPOKE_2_IKE_GW local-identity distinguished-name
set security ike gateway SPOKE_2_IKE_GW remote-identity distinguished-name
set security ike gateway SPOKE_2_IKE_GW ppk-profile SPOKE_2_KM_PROFILE_1
set security ike gateway SPOKE_2_IKE_GW version v2-only
set security ipsec proposal SPOKE_2_IPSEC_PROP protocol esp
set security ipsec proposal SPOKE_2_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal SPOKE_2_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec policy SPOKE_2_IPSEC_POL proposals SPOKE_2_IPSEC_PROP
set security ipsec vpn SPOKE_2_IPSEC_VPN bind-interface st0.2
set security ipsec vpn SPOKE_2_IPSEC_VPN ike gateway SPOKE_2_IKE_GW
set security ipsec vpn SPOKE_2_IPSEC_VPN ike ipsec-policy SPOKE_2_IPSEC_POL
set security ipsec vpn SPOKE_2_IPSEC_VPN traffic-selector ts1 local-ip 192.168.70.0/24
set security ipsec vpn SPOKE_2_IPSEC_VPN traffic-selector ts1 remote-ip 192.168.90.0/24
set interfaces ge-0/0/2 unit 0 family inet address 172.18.10.3/24
set interfaces ge-0/0/1 unit 0 family inet address 192.168.70.1/24
set interfaces st0 unit 2 family inet
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone vpn interfaces st0.2
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
set security policies from-zone trust to-zone vpn policy vpn_out match source-address any
set security policies from-zone trust to-zone vpn policy vpn_out match destination-address any
set security policies from-zone trust to-zone vpn policy vpn_out match application any
set security policies from-zone trust to-zone vpn policy vpn_out then permit
set security policies from-zone vpn to-zone trust policy vpn_in match source-address any
set security policies from-zone vpn to-zone trust policy vpn_in match destination-address any
set security policies from-zone vpn to-zone trust policy vpn_in match application any
set security policies from-zone vpn to-zone trust policy vpn_in then permit
```

15

CHAPTER

Monitoring VPN

Monitoring VPN Traffic

IN THIS SECTION

VPN monitoring enables you to determine the reachability of peer devices by sending Internet Control Message Protocol (ICMP) requests to the peers.

Understanding VPN Alarms and Auditing

Configure the following command to enable security event logging during the initial set up of the device. This feature is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 devices and vSRX Virtual Firewall instances.

```
set security log cache
```

The administrators (audit, cryptographic, IDS and security) cannot modify the security event logging configuration if the above command is configured and each administrator role is configured to have a distinct, unique set of privileges apart from all other administrative roles.

Alarms are triggered by a VPN failure. A VPN alarm is generated when the system monitors any of the following audited events:

- `Authentication failures`—You can configure the device to generate a system alarm when the packet authentication failures reaches a specified number.
- `Encryption and decryption failures`—You can configure the device to generate a system alarm when encryption or decryption failures exceed a specified number.
- `IKE Phase 1 and IKE Phase 2 failures`—Internet Key Exchange (IKE) Phase 1 negotiations are used to establish IKE security associations (SAs). These SAs protect the IKE Phase 2 negotiations. You can

configure the device to generate a system alarm when IKE Phase 1 or IKE Phase 2 failures exceed a specified number.

- `Self-test failures`—Self-tests are tests that a device runs upon power on or reboot to verify whether security software is implemented correctly on your device.

  Self-tests ensure the correctness of cryptographic algorithms. The Junos-FIPS image performs self-tests automatically upon power-on, and continuously for key-pair generation. In either domestic or FIPS images, self-tests can be configured to be performed according to a defined schedule, upon demand or immediately after key generation.

  You can configure the device to generate a system alarm when a self-test failure occurs.

- `IDP flow policy attacks`—An intrusion detection and prevention (IDP) policy allows you to enforce various attack detection and prevention techniques on network traffic. You can configure the device to generate a system alarm when IDP flow policy violations occur.

- `Replay attacks`—A replay attack is a network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. You can configure the device to generate a system alarm when a replay attack occurs.

The syslog messages are included in the following cases:

- Failed symmetric key generation
- Failed asymmetric key generation
- Failed manual key distribution
- Failed automated key distribution
- Failed key destruction
- Failed key handling and storage
- Failed data encryption or decryption
- Failed signature
- Failed key agreement
- Failed cryptographic hashing
- IKE failure
- Failed authentication of the received packets
- Decryption error due to invalid padding content

- Mismatch in the length specified in the alternative subject field of the certificate received from a remote VPN peer device.

Alarms are raised based on syslog messages. Every failure is logged, but an alarm is generated only when a threshold is reached.

To view the alarm information, run the `show security alarms` command. The violation count and the alarm do not persist across system reboots. After a reboot, the violation count resets to zero, and the alarm is cleared from the alarm queue.

After appropriate actions have been taken, you can clear the alarm. The alarm remains in the queue until you clear it (or until you reboot the device). To clear the alarm, run the `clear security alarms` command.

SEE ALSO

IPsec Overview | 20

IPsec VPN Topologies on SRX Series Firewalls | 166

Understanding VPN Monitoring

IN THIS SECTION

VPN monitoring uses *ICMP* echo requests (or *pings*) to determine if a VPN tunnel is up. When VPN monitoring is enabled, the security device sends pings through the VPN tunnel to the peer gateway or to a specified destination at the other end of the tunnel. Pings are sent by default at intervals of 10 seconds for up to 10 consecutive times. If no reply is received after 10 consecutive pings, the VPN is considered to be down and the IPsec security association (*SA*) is cleared.

VPN monitoring is enabled for a specified VPN by configuring the `vpn-monitor` option at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level. The peer gateway's IP address is the default destination; however, you can specify a different destination IP address (such as a server) at the other end of the tunnel. The local *tunnel endpoint* is the default source interface, but you can specify a different interface name.

VPN monitoring of an externally connected device (such as a PC) is not supported on SRX5400, SRX5600, and SRX5800 devices. The destination for VPN monitoring must be a local interface on the SRX5400, SRX5600, or SRX5800 device.

The VPN monitoring `optimized` option sends pings only when there is outgoing traffic and no incoming traffic through the VPN tunnel. If there is incoming traffic through the VPN tunnel, the security device considers the tunnel to be active and does not send pings to the peer. Configuring the `optimized` option can save resources on the security device because pings are only sent when peer liveliness needs to be determined. Sending pings can also activate costly backup links that would otherwise not be used.

You can configure the interval at which pings are sent and the number of consecutive pings that are sent without a reply before the VPN is considered to be down. These are configured with the `interval` and `threshold` options, respectively, at the [`edit security ipsec vpn-monitor-options`] hierarchy level.

VPN monitoring can cause tunnel flapping in some VPN environments if ping packets are not accepted by the peer based on the packet's source or destination IP address.

Understanding IPsec Datapath Verification

Overview

By default, the state of the secure tunnel (st0) interfaces configured in point-to-point mode in route-based VPNs is based on the state of the VPN tunnel. Soon after the IPsec SA is established, routes associated with the st0 interface are installed in the Junos OS forwarding table. In certain network topologies, such as where a transit firewall is located between the VPN tunnel endpoints, IPsec data traffic that uses active routes for an established VPN tunnel on the st0 interface may be blocked by the transit firewall. This can result in traffic loss.

When you enable the IPsec datapath verification, the st0 interface is not brought up and activated until the datapath is verified. The verification is configured with the `set security ipsec vpn` *`vpn-name`* `vpn-monitor verify-path` statement for route-based, site-to-site, and dynamic endpoint VPN tunnels.

If there is a NAT device in front of the peer tunnel endpoint, the IP address of the peer tunnel endpoint is translated to the IP address of the NAT device. For the VPN monitor ICMP request to reach the peer tunnel endpoint, you need to explicitly specify the original, untranslated IP address of the peer tunnel endpoint behind the NAT device. This is configured with the `set security ipsec vpn` *`vpn-name`* `vpn-monitor verify-path destination-ip` configuration.

Starting in Junos OS Release 15.1X49-D120, you can configure the size of the packet that is used to verify an IPsec datapath before the `st0` interface is brought up. Use the `set security ipsec vpn vpn-name vpn-monitor verify-path packet-size` configuration. The configurable packet size ranges from 64 to 1350 bytes; the default is 64 bytes.

Caveats

The source interface and destination IP addresses that can be configured for VPN monitor operation have no effect on the IPsec datapath verification. The source for the ICMP requests in the IPsec datapath verification is the local tunnel endpoint.

When you enable IPsec datapath verification, VPN monitoring is automatically activated and used after the st0 interface is brought up. We recommend that you configure the VPN monitor optimized option with the `set security ipsec vpn *vpn-name* vpn-monitor optimized` command whenever you enable IPsec datapath verification.

If a chassis cluster failover occurs during the IPsec datapath verification, the new active node starts the verification again. The st0 interface is not activated until the verification succeeds.

No IPsec datapath verification is performed for IPsec SA rekeys, because the st0 interface state does not change for rekeys.

IPsec datapath verification is not supported on st0 interfaces configured in point-to-multipoint mode that are used with AutoVPN, Auto Discovery VPN, and multiple traffic selectors. VPN monitoring and IPsec datapath verification do not support IPv6 addresses, so IPsec datapath verification cannot be used with IPv6 tunnels.

Understanding Global SPI and VPN Monitoring Features

You can monitor and maintain the efficient operation of your VPN using the following global VPN features:

- SPI—Peers in a security association (SA) can become unsynchronized when one of the peers fails. For example, if one of the peers reboots, it might send an incorrect security parameter index (SPI). You can enable the device to detect such an event and resynchronize the peers by configuring the bad SPI response feature.

- VPN monitoring—You can use the global VPN monitoring feature to periodically send Internet Control Message Protocol (ICMP) requests to the peer to determine if the peer is reachable.

Understanding VPN Monitoring and DPD

VPN monitoring and dead peer detection (DPD) are features available on SRX Series Firewalls to verify the availability of VPN peer devices. This section compares the operation and configuration of these features.

The SRX Series Firewall responds to DPD messages sent by VPN peers even if DPD is not configured on the device. You can configure the SRX Series Firewall to initiate DPD messages to VPN peers. You can also configure DPD and VPN monitoring to operate simultaneously on the same SRX Series Firewall, although the number of peers that can be monitored with either method is reduced.

VPN monitoring is a Junos OS mechanism that monitors only Phase 2 security associations (SAs). VPN monitoring is enabled on a per-VPN basis with the `vpn-monitor` statement at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level. The destination IP and source interface must be specified. The `optimized` option enables the device to use traffic patterns as evidence of peer liveliness; ICMP requests are suppressed.

VPN monitoring options are configured with the `vpn-monitor-options` statement at the [`edit security ipsec`] hierarchy level. These options apply to all VPNs for which VPN monitoring is enabled. Options you can configure include the interval at which ICMP requests are sent to the peer (the default is 10 seconds) and the number of consecutive ICMP requests sent without receiving a response before the peer is considered unreachable (the default is 10 consecutive requests).

DPD is an implementation of RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*. It operates at the IKE level and monitors the peer based on both IKE and IPsec traffic activity.

DPD is configured on an individual IKE gateway with the `dead-peer-detection` statement at the [`edit security ike gateway` *`gateway-name`*] hierarchy level. You can configure DPD modes of operation. The default (optimized) mode sends DPD messages to the peer if there is no incoming IKE or IPsec traffic within a configured interval after the local device sends outgoing packets to the peer. Other configurable options include the interval at which DPD messages are sent to the peer (the default is 10 seconds) and the number of consecutive DPD messages sent without receiving a response before the peer is considered unavailable (the default is five consecutive requests).

Understanding Dead Peer Detection

Dead peer detection (DPD) is a method that network devices use to verify the current existence and availability of other peer devices.

You can use DPD as an alternative to VPN monitoring. VPN monitoring applies to an individual IPsec VPN, while DPD is configured only in an individual IKE gateway context.

A device performs DPD verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE messages) to a peer and waiting for DPD acknowledgments (R-U-THERE-ACK messages) from the peer. The device sends an R-U-THERE message only if it has not received any traffic from the peer during a specified DPD interval. If the device receives an R-U-THERE-ACK message from the peer during this interval, it considers the peer alive. If the device receives traffic on the tunnel from the peer, it resets its R-U-THERE message counter for that tunnel, thus starting a new interval. If the device does not receive an R-U-THERE-ACK message during the interval, it considers the peer dead. When the device changes the status of a peer device to be dead, the device removes the Phase 1 security association (SA) and all Phase 2 SAs for that peer.

The following DPD modes are supported on the SRX Series Firewalls:

- Optimized—R-U-THERE messages are triggered if there is no incoming IKE or IPsec traffic within a configured interval after the device sends outgoing packets to the peer. This is the default mode.

- Probe idle tunnel—R-U-THERE messages are triggered if there is no incoming or outgoing IKE or IPsec traffic within a configured interval. R-U-THERE messages are sent periodically to the peer until there is traffic activity. This mode helps in early detection of a downed peer and makes the tunnel available for data traffic.

  When multiple traffic selectors are configured for a VPN, multiple tunnels can be established for the same IKE SA. In this scenario, the probe idle tunnel mode triggers R-U-THERE messages to be sent if any tunnel associated with the IKE SA becomes idle, even though there may be traffic in another tunnel for the same IKE SA.

- Always send—R-U-THERE messages are sent at configured intervals regardless of traffic activity between the peers.

  We recommend that the probe idle tunnel mode be used instead of the `always-send` mode.

DPD timers are active as soon as the Phase 1 SA is established. The DPD behavior is the same for both IKEv1 and IKEv2 protocols.

You can configure the following DPD parameters:

- The interval parameter specifies the amount of time (expressed in seconds) the device waits for traffic from its peer before sending an R-U-THERE message. The default interval is 10 seconds. Starting with Junos OS Release 15.1X49-D130, the permissible interval parameter range at which R-U-THERE messages are sent to the peer device is reduced from 10 through 60 seconds to 2 seconds through 60 seconds. The minimum threshold parameter should be 3, when the DPD interval parameter is set less than 10 seconds.

- The threshold parameter specifies the maximum number of times to send the R-U-THERE message without a response from the peer before considering the peer dead. The default number of transmissions is five times, with a permissible range of 1 to 5 retries.

Note the following considerations before configuring DPD:

- When a DPD configuration is added to an existing gateway with active tunnels, R-U-THERE messages are started without clearing Phase 1 or Phase 2 SAs.

- When a DPD configuration is deleted from an existing gateway with active tunnels, R-U-THERE messages are stopped for the tunnels. IKE and IPsec SAs are not affected.

- Modifying any DPD configuration option such as the mode, interval, or threshold values updates the DPD operation without clearing Phase 1 or Phase 2 SAs.

- If the IKE gateway is configured with DPD and VPN monitoring but the option to establish tunnels immediately is not configured, DPD does not initiate Phase 1 negotiation. When DPD is configured, the establish tunnels immediately option must also be configured at the same time to tear down the st0 interface when there are no phase 1 and phase 2 SAs available.

- If the IKE gateway is configured with multiple peer IP addresses and DPD but Phase 1 SA fails to be established to the first peer IP address, a Phase 1 SA is attempted with the next peer IP address. DPD is active only after a Phase 1 SA is established.

- If the IKE gateway is configured with multiple peer IP addresses and DPD but DPD fails with the current peer's IP address, the Phase 1 and Phase 2 SAs are cleared and a failover to the next peer IP address is triggered.

- More than one Phase 1 or Phase 2 SA can exist with the same peer because of simultaneous negotiations. In this case, R-U-THERE messages are sent on all Phase 1 SAs. Failure to receive DPD responses for the configured number of consecutive times clears the Phase 1 SA and the associated Phase 2 SA (for IKEv2 only).

SEE ALSO

verify-path | **1660**

IPsec Overview | **20**

Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device | **631**

Understanding Tunnel Events

When there is a network problem related to a VPN, after the tunnel comes up only the tunnel status is tracked. Many issues can occur before the tunnel comes up. Hence, instead of tracking only the tunnel status, tunnel down issues, or negotiation failures, successful events such as successful IPsec SA negotiations, IPsec rekey, and IKE SA rekeys are now tracked. These events are called tunnel events.

For Phase 1 and Phase 2, negotiation events for a given tunnel are tracked along with the events that occur in external daemons like AUTHD or PKID. When a tunnel event occurs multiple times, only one entry is maintained with the updated time and the number of times that event occurred.

Overall, 16 events are tracked: eight events for Phase 1 and eight events for Phase 2. Some events can reoccur and fill up the event memory, resulting in important events being removed. To avoid overwriting, an event is not stored unless a tunnel is down.

The following special events fall into this category:

- Lifetime in kilobytes expired for IPsec SA

- Hard lifetime of IPsec SA expired

- IPsec SA delete payload received from peer, corresponding IPsec SAs cleared

- Cleared unused redundant backup IPsec SA pairs
- IPsec SAs cleared as corresponding IKE SA deleted

AutoVPN tunnels are created and removed dynamically and consequently tunnel events corresponding to these tunnels are short lived. Sometimes these tunnel events cannot be associated with any tunnel so system logging is used for debugging instead.

SEE ALSO

IPsec Overview | 20

Example: Setting an Audible Alert as Notification of a Security Alarm

IN THIS SECTION

This example shows how to configure a device to generate a system alert beep when a new security event occurs. By default, alarms are not audible. This feature is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 devices and vSRX Virtual Firewall instances.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

In this example, you set an audible beep to be generated in response to a security alarm.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To set an audible alarm:

1. Enable security alarms.

```
[edit]
user@host# edit security alarms
```

2. Specify that you want to be notified of security alarms with an audible beep.

```
[edit security alarms]
user@host# set audible
```

3. If you are done configuring the device, commit the configuration.

```
[edit security alarms]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the `show security alarms detail` command.

SEE ALSO

IPsec Overview | 20

Example: Generating Security Alarms in Response to Potential Violations

IN THIS SECTION

This example shows how to configure the device to generate a system alarm when a potential violation occurs. By default, no alarm is raised when a potential violation occurs. This feature is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, and SRX1500 devices and vSRX Virtual Firewall instances.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

In this example, you configure an alarm to be raised when:

- The number of authentication failures exceeds 6.
- The cryptographic self-test fails.
- The non-cryptographic self-test fails.
- The key generation self-test fails.
- The number of encryption failures exceeds 10.
- The number of decryption failures exceeds 1.
- The number of IKE Phase 1 failures exceeds 10.
- The number of IKE Phase 2 failure exceeds 1.
- A replay attack occurs.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security alarms potential-violation authentication 6
set security alarms potential-violation cryptographic-self-test
set security alarms potential-violation non-cryptographic-self-test
set security alarms potential-violation key-generation-self-test
set security alarms potential-violation encryption-failures threshold 10
set security alarms potential-violation decryption-failures threshold 1
set security alarms potential-violation ike-phase1-failures threshold 10
set security alarms potential-violation ike-phase2-failures threshold 1
set security alarms potential-violation replay-attacks
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure alarms in response to potential violations:

1. Enable security alarms.

```
[edit]
user@host# edit security alarms
```

2. Specify that an alarm should be raised when an authentication failure occurs.

```
[edit security alarms potential-violation]
user@host# set authentication 6
```

3. Specify that an alarm should be raised when a cryptographic self-test failure occurs.

```
[edit security alarms potential-violation]
user@host# set cryptographic-self-test
```

4. Specify that an alarm should be raised when a non-cryptographic self-test failure occurs.

```
[edit security alarms potential-violation]
user@host# set non-cryptographic-self-test
```

5. Specify that an alarm should be raised when a key generation self-test failure occurs.

```
[edit security alarms potential-violation]
user@host# set key-generation-self-test
```

6. Specify that an alarm should be raised when an encryption failure occurs.

```
[edit security alarms potential-violation]
user@host# set encryption-failures threshold 10
```

7. Specify that an alarm should be raised when a decryption failure occurs.

```
[edit security alarms potential-violation]
user@host# set decryption-failures threshold 1
```

8. Specify that an alarm should be raised when an IKE Phase 1 failure occurs.

```
[edit security alarms potential-violation]
user@host# set ike-phase1-failures threshold 10
```

9. Specify that an alarm should be raised when an IKE Phase 2 failure occurs.

   ```
   [edit security alarms potential-violation]
   user@host# set ike-phase2-failures threshold 1
   ```

10. Specify that an alarm should be raised when a replay attack occurs.

    ```
    [edit security alarms potential-violation]
    user@host# set replay-attacks
    ```

Results

From configuration mode, confirm your configuration by entering the `show security alarms` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
potential-violation {
    authentication 6;
    cryptographic-self-test;
    decryption-failures {
        threshold 1;
    }
    encryption-failures {
        threshold 10;
    }
    ike-phase1-failures {
        threshold 10;
    }
    ike-phase2-failures {
        threshold 1;
    }
    key-generation-self-test;
    non-cryptographic-self-test;
    replay-attacks;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

To confirm that the configuration is working properly, from operational mode, enter the `show security alarms` command.

SEE ALSO

VPN Support for Inserting Services Processing Cards | 175

Release History Table

Release	Description
15.1X49-D130	Starting with Junos OS Release 15.1X49-D130, the permissible interval parameter range at which R-U-THERE messages are sent to the peer device is reduced from 10 through 60 seconds to 2 seconds through 60 seconds. The minimum threshold parameter should be 3, when the DPD interval parameter is set less than 10 seconds.
15.1X49-D120	Starting in Junos OS Release 15.1X49-D120, you can configure the size of the packet that is used to verify an IPsec datapath before the `st0` interface is brought up.

RELATED DOCUMENTATION

IPsec VPN Overview | 165

Monitoring IPsec VPN Sessions

IN THIS SECTION

Understanding VPN Monitoring

IN THIS SECTION

VPN monitoring uses *ICMP* echo requests (or *pings*) to determine if a VPN tunnel is up. When VPN monitoring is enabled, the security device sends pings through the VPN tunnel to the peer gateway or to a specified destination at the other end of the tunnel. Pings are sent by default at intervals of 10 seconds for up to 10 consecutive times. If no reply is received after 10 consecutive pings, the VPN is considered to be down and the IPsec security association (*SA*) is cleared.

VPN monitoring is enabled for a specified VPN by configuring the `vpn-monitor` option at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level. The peer gateway's IP address is the default destination; however, you can specify a different destination IP address (such as a server) at the other end of the tunnel. The local *tunnel endpoint* is the default source interface, but you can specify a different interface name.

VPN monitoring of an externally connected device (such as a PC) is not supported on SRX5400, SRX5600, and SRX5800 devices. The destination for VPN monitoring must be a local interface on the SRX5400, SRX5600, or SRX5800 device.

The VPN monitoring `optimized` option sends pings only when there is outgoing traffic and no incoming traffic through the VPN tunnel. If there is incoming traffic through the VPN tunnel, the security device considers the tunnel to be active and does not send pings to the peer. Configuring the `optimized` option can save resources on the security device because pings are only sent when peer liveliness needs to be determined. Sending pings can also activate costly backup links that would otherwise not be used.

You can configure the interval at which pings are sent and the number of consecutive pings that are sent without a reply before the VPN is considered to be down. These are configured with the `interval` and `threshold` options, respectively, at the [`edit security ipsec vpn-monitor-options`] hierarchy level.

VPN monitoring can cause tunnel flapping in some VPN environments if ping packets are not accepted by the peer based on the packet's source or destination IP address.

Understanding IPsec Datapath Verification

Overview

By default, the state of the secure tunnel (st0) interfaces configured in point-to-point mode in route-based VPNs is based on the state of the VPN tunnel. Soon after the IPsec SA is established, routes associated with the st0 interface are installed in the Junos OS forwarding table. In certain network topologies, such as where a transit firewall is located between the VPN tunnel endpoints, IPsec data traffic that uses active routes for an established VPN tunnel on the st0 interface may be blocked by the transit firewall. This can result in traffic loss.

When you enable the IPsec datapath verification, the st0 interface is not brought up and activated until the datapath is verified. The verification is configured with the `set security ipsec vpn` *`vpn-name`* `vpn-monitor verify-path` statement for route-based, site-to-site, and dynamic endpoint VPN tunnels.

If there is a NAT device in front of the peer tunnel endpoint, the IP address of the peer tunnel endpoint is translated to the IP address of the NAT device. For the VPN monitor ICMP request to reach the peer tunnel endpoint, you need to explicitly specify the original, untranslated IP address of the peer tunnel endpoint behind the NAT device. This is configured with the `set security ipsec vpn` *`vpn-name`* `vpn-monitor verify-path destination-ip` configuration.

Starting in Junos OS Release 15.1X49-D120, you can configure the size of the packet that is used to verify an IPsec datapath before the `st0` interface is brought up. Use the `set security ipsec vpn vpn-name vpn-monitor verify-path packet-size` configuration. The configurable packet size ranges from 64 to 1350 bytes; the default is 64 bytes.

Caveats

The source interface and destination IP addresses that can be configured for VPN monitor operation have no effect on the IPsec datapath verification. The source for the ICMP requests in the IPsec datapath verification is the local tunnel endpoint.

When you enable IPsec datapath verification, VPN monitoring is automatically activated and used after the st0 interface is brought up. We recommend that you configure the VPN monitor optimized option with the `set security ipsec vpn` *`vpn-name`* `vpn-monitor optimized` command whenever you enable IPsec datapath verification.

If a chassis cluster failover occurs during the IPsec datapath verification, the new active node starts the verification again. The st0 interface is not activated until the verification succeeds.

No IPsec datapath verification is performed for IPsec SA rekeys, because the st0 interface state does not change for rekeys.

IPsec datapath verification is not supported on st0 interfaces configured in point-to-multipoint mode that are used with AutoVPN, Auto Discovery VPN, and multiple traffic selectors. VPN monitoring and

IPsec datapath verification do not support IPv6 addresses, so IPsec datapath verification cannot be used with IPv6 tunnels.

Understanding Global SPI and VPN Monitoring Features

You can monitor and maintain the efficient operation of your VPN using the following global VPN features:

- SPI—Peers in a security association (SA) can become unsynchronized when one of the peers fails. For example, if one of the peers reboots, it might send an incorrect security parameter index (SPI). You can enable the device to detect such an event and resynchronize the peers by configuring the bad SPI response feature.

- VPN monitoring—You can use the global VPN monitoring feature to periodically send Internet Control Message Protocol (ICMP) requests to the peer to determine if the peer is reachable.

Understanding VPN Monitoring and DPD

VPN monitoring and dead peer detection (DPD) are features available on SRX Series Firewalls to verify the availability of VPN peer devices. This section compares the operation and configuration of these features.

The SRX Series Firewall responds to DPD messages sent by VPN peers even if DPD is not configured on the device. You can configure the SRX Series Firewall to initiate DPD messages to VPN peers. You can also configure DPD and VPN monitoring to operate simultaneously on the same SRX Series Firewall, although the number of peers that can be monitored with either method is reduced.

VPN monitoring is a Junos OS mechanism that monitors only Phase 2 security associations (SAs). VPN monitoring is enabled on a per-VPN basis with the `vpn-monitor` statement at the [`edit security ipsec vpn` *`vpn-name`*] hierarchy level. The destination IP and source interface must be specified. The `optimized` option enables the device to use traffic patterns as evidence of peer liveliness; ICMP requests are suppressed.

VPN monitoring options are configured with the `vpn-monitor-options` statement at the [`edit security ipsec`] hierarchy level. These options apply to all VPNs for which VPN monitoring is enabled. Options you can configure include the interval at which ICMP requests are sent to the peer (the default is 10 seconds) and the number of consecutive ICMP requests sent without receiving a response before the peer is considered unreachable (the default is 10 consecutive requests).

DPD is an implementation of RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*. It operates at the IKE level and monitors the peer based on both IKE and IPsec traffic activity.

DPD is configured on an individual IKE gateway with the `dead-peer-detection` statement at the [`edit security ike gateway` *`gateway-name`*] hierarchy level. You can configure DPD modes of operation. The default (optimized) mode sends DPD messages to the peer if there is no incoming IKE or IPsec traffic within a configured interval after the local device sends outgoing packets to the peer. Other configurable options include the interval at which DPD messages are sent to the peer (the default is 10 seconds) and the number of consecutive DPD messages sent without receiving a response before the peer is considered unavailable (the default is five consecutive requests).

Understanding Dead Peer Detection

Dead peer detection (DPD) is a method that network devices use to verify the current existence and availability of other peer devices.

You can use DPD as an alternative to VPN monitoring. VPN monitoring applies to an individual IPsec VPN, while DPD is configured only in an individual IKE gateway context.

A device performs DPD verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE messages) to a peer and waiting for DPD acknowledgments (R-U-THERE-ACK messages) from the peer. The device sends an R-U-THERE message only if it has not received any traffic from the peer during a specified DPD interval. If the device receives an R-U-THERE-ACK message from the peer during this interval, it considers the peer alive. If the device receives traffic on the tunnel from the peer, it resets its R-U-THERE message counter for that tunnel, thus starting a new interval. If the device does not receive an R-U-THERE-ACK message during the interval, it considers the peer dead. When the device changes the status of a peer device to be dead, the device removes the Phase 1 security association (SA) and all Phase 2 SAs for that peer.

The following DPD modes are supported on the SRX Series Firewalls:

- Optimized—R-U-THERE messages are triggered if there is no incoming IKE or IPsec traffic within a configured interval after the device sends outgoing packets to the peer. This is the default mode.
- Probe idle tunnel—R-U-THERE messages are triggered if there is no incoming or outgoing IKE or IPsec traffic within a configured interval. R-U-THERE messages are sent periodically to the peer until there is traffic activity. This mode helps in early detection of a downed peer and makes the tunnel available for data traffic.

  When multiple traffic selectors are configured for a VPN, multiple tunnels can be established for the same IKE SA. In this scenario, the probe idle tunnel mode triggers R-U-THERE messages to be sent if any tunnel associated with the IKE SA becomes idle, even though there may be traffic in another tunnel for the same IKE SA.
- Always send—R-U-THERE messages are sent at configured intervals regardless of traffic activity between the peers.

  We recommend that the probe idle tunnel mode be used instead of the `always-send` mode.

DPD timers are active as soon as the Phase 1 SA is established. The DPD behavior is the same for both IKEv1 and IKEv2 protocols.

You can configure the following DPD parameters:

- The interval parameter specifies the amount of time (expressed in seconds) the device waits for traffic from its peer before sending an R-U-THERE message. The default interval is 10 seconds. Starting with Junos OS Release 15.1X49-D130, the permissible interval parameter range at which R-U-THERE messages are sent to the peer device is reduced from 10 through 60 seconds to 2 seconds through 60 seconds. The minimum threshold parameter should be 3, when the DPD interval parameter is set less than 10 seconds.

- The threshold parameter specifies the maximum number of times to send the R-U-THERE message without a response from the peer before considering the peer dead. The default number of transmissions is five times, with a permissible range of 1 to 5 retries.

Note the following considerations before configuring DPD:

- When a DPD configuration is added to an existing gateway with active tunnels, R-U-THERE messages are started without clearing Phase 1 or Phase 2 SAs.

- When a DPD configuration is deleted from an existing gateway with active tunnels, R-U-THERE messages are stopped for the tunnels. IKE and IPsec SAs are not affected.

- Modifying any DPD configuration option such as the mode, interval, or threshold values updates the DPD operation without clearing Phase 1 or Phase 2 SAs.

- If the IKE gateway is configured with DPD and VPN monitoring but the option to establish tunnels immediately is not configured, DPD does not initiate Phase 1 negotiation. When DPD is configured, the establish tunnels immediately option must also be configured at the same time to tear down the st0 interface when there are no phase 1 and phase 2 SAs available.

- If the IKE gateway is configured with multiple peer IP addresses and DPD but Phase 1 SA fails to be established to the first peer IP address, a Phase 1 SA is attempted with the next peer IP address. DPD is active only after a Phase 1 SA is established.

- If the IKE gateway is configured with multiple peer IP addresses and DPD but DPD fails with the current peer's IP address, the Phase 1 and Phase 2 SAs are cleared and a failover to the next peer IP address is triggered.

- More than one Phase 1 or Phase 2 SA can exist with the same peer because of simultaneous negotiations. In this case, R-U-THERE messages are sent on all Phase 1 SAs. Failure to receive DPD responses for the configured number of consecutive times clears the Phase 1 SA and the associated Phase 2 SA (for IKEv2 only).

SEE ALSO

verify-path | **1660**

IPsec Overview | 20

Example: Configuring a Policy-Based VPN with Both an Initiator and a Responder Behind a NAT Device | 631

Understanding Tunnel Events

When there is a network problem related to a VPN, after the tunnel comes up only the tunnel status is tracked. Many issues can occur before the tunnel comes up. Hence, instead of tracking only the tunnel status, tunnel down issues, or negotiation failures, successful events such as successful IPsec SA negotiations, IPsec rekey, and IKE SA rekeys are now tracked. These events are called tunnel events.

For Phase 1 and Phase 2, negotiation events for a given tunnel are tracked along with the events that occur in external daemons like AUTHD or PKID. When a tunnel event occurs multiple times, only one entry is maintained with the updated time and the number of times that event occurred.

Overall, 16 events are tracked: eight events for Phase 1 and eight events for Phase 2. Some events can reoccur and fill up the event memory, resulting in important events being removed. To avoid overwriting, an event is not stored unless a tunnel is down.

The following special events fall into this category:

- Lifetime in kilobytes expired for IPsec SA
- Hard lifetime of IPsec SA expired
- IPsec SA delete payload received from peer, corresponding IPsec SAs cleared
- Cleared unused redundant backup IPsec SA pairs
- IPsec SAs cleared as corresponding IKE SA deleted

AutoVPN tunnels are created and removed dynamically and consequently tunnel events corresponding to these tunnels are short lived. Sometimes these tunnel events cannot be associated with any tunnel so system logging is used for debugging instead.

SEE ALSO

IPsec Overview | 20

Release History Table

Release	Description
15.1X49-D130	Starting with Junos OS Release 15.1X49-D130, the permissible interval parameter range at which R-U-THERE messages are sent to the peer device is reduced from 10 through 60 seconds to 2 seconds through 60 seconds. The minimum threshold parameter should be 3, when the DPD interval parameter is set less than 10 seconds.
15.1X49-D120	Starting in Junos OS Release 15.1X49-D120, you can configure the size of the packet that is used to verify an IPsec datapath before the `st0` interface is brought up.

16

CHAPTER

Performance Tuning

VPN Session Affinity

IN THIS SECTION

The performance of IPsec VPN traffic to minimize packet forwarding overhead can be optimized by enabling VPN session affinity and performance acceleration.

Understanding VPN Session Affinity

VPN session affinity occurs when a cleartext session is located in a Services Processing Unit (SPU) that is different from the SPU where the IPsec tunnel session is located. The goal of VPN session affinity is to locate the cleartext and IPsec tunnel session in the same SPU. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices.

Without VPN session affinity, a cleartext session created by a flow might be located in one SPU and the tunnel session created by IPsec might be located in another SPU. An SPU to SPU forward or hop is needed to route cleartext packets to the IPsec tunnel.

By default, VPN session affinity is disabled on SRX Series Firewalls. When VPN session affinity is enabled, a new cleartext session is placed on the same SPU as the IPsec tunnel session. Existing cleartext sessions are not affected.

Junos OS Release 15.1X49-D10 introduces the SRX5K-MPC3-100G10G (IOC3) and the SRX5K-MPC3-40G10G (IOC3) for SRX5400, SRX5600, and SRX5800 devices.

The SRX5K-MPC (IOC2) and the IOC3 support VPN session affinity through improved flow module and session cache. With IOCs, the flow module creates sessions for IPsec tunnel-based traffic before encryption and after decryption on its tunnel-anchored SPU and installs the session cache for the sessions so that the IOC can redirect the packets to the same SPU to minimize packet forwarding overhead. Express Path (previously known as services offloading) traffic and NP cache traffic share the same session cache table on the IOCs.

To display active tunnel sessions on SPUs, use the `show security ipsec security-association` command and specify the Flexible PIC Concentrator (FPC) and *Physical Interface Card* (PIC) slots that contain the SPU. For example:

```
user@host> show security ipsec security-association fpc 3 pic 0
  Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon vsys Port  Gateway
  <131073 ESP:aes-128/sha1 18c4fd00 491/  128000 - root 500   203.0.113.11
  >131073 ESP:aes-128/sha1 188c0750 491/  128000 - root 500   203.0.113.11
```

You need to evaluate the tunnel distribution and traffic patterns in your network to determine if VPN session affinity should be enabled.

Starting with Junos OS Release 12.3X48-D50, Junos OS Release 15.1X49-D90, and Junos OS Release 17.3R1, if VPN session affinity is enabled on SRX5400, SRX5600, and SRX5800 devices, the tunnel overhead is calculated according to the negotiated encryption and authentication algorithms on the anchor Services Processing Unit (SPU). If the configured encryption or authentication changes, the tunnel overhead is updated on the anchor SPU when a new IPsec security association is established.

The VPN session affinity limitations are as follows:

- Traffic across logical systems is not supported.
- If there is a route change, established cleartext sessions remain on an SPU and traffic is rerouted if possible. Sessions created after the route change can be set up on a different SPU.
- VPN session affinity only affects self traffic that terminates on the device (also known as host-inbound traffic); self traffic that originates from the device (also known as host-outbound traffic) is not affected.
- Multicast replication and forwarding performance is not affected.

SEE ALSO

- Understanding Traffic Processing on SRX5000 Line Devices
- Understanding Session Cache
- Express Path Overview
- Example: Enabling Express Path in Security Policies
- Express Path

Enabling VPN Session Affinity

By default, VPN session affinity is disabled on SRX Series Firewalls. Enabling VPN session affinity can improve VPN throughput under certain conditions. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices. This section describes how to use the CLI to enable VPN session affinity.

Determine if clear-text sessions are being forwarded to IPsec tunnel sessions on a different SPU. Use the `show security flow session` command to display session information about clear-text sessions.

```
user@host> show security flow session
Flow Sessions on FPC3 PIC0:

Session ID: 60000001, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/6204 --> 203.0.113.6/41264;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 60000002, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 60000003, Policy name: self-traffic-policy/1, Timeout: 58, Valid
  In: 203.0.113.6/500 --> 203.0.113.11/500;udp, If: .local..0, Pkts: 105386, Bytes: 12026528
  Out: 203.0.113.11/500 --> 203.0.113.6/500;udp, If: ge-0/0/2.0, Pkts: 106462, Bytes: 12105912

Session ID: 60017354, Policy name: N/A, Timeout: 1784, Valid
  In: 0.0.0.0/0 --> 0.0.0.0/0;0, If: N/A, Pkts: 0, Bytes: 0
  Out: 198.51.100.156/23 --> 192.0.2.155/53051;tcp, If: N/A, Pkts: 0, Bytes: 0
Total sessions: 4

Flow Sessions on FPC6 PIC0:

Session ID: 120000001, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 120000002, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 120031730, Policy name: default-policy-00/2, Timeout: 1764, Valid
  In: 192.0.2.155/53051 --> 198.51.100.156/23;tcp, If: ge-0/0/1.0, Pkts: 44, Bytes: 2399
  Out: 198.51.100.156/23 --> 192.0.2.155/53051;tcp, If: st0.0, Pkts: 35, Bytes: 2449
Total sessions: 3
```

In the example, there is a tunnel session on FPC 3, PIC 0 and a clear-text session on FPC 6, PIC 0. A forwarding session (session ID 60017354) is set up on FPC 3, PIC 0.

Junos OS Release 15.1X49-D10 introduces session affinity support on IOCs (SRX5K-MPC [IOC2], SRX5K-MPC3-100G10G [IOC3], and SRX5K-MPC3-40G10G [IOC3]) and Junos OS Release 12.3X48-D30 introduces session affinity support on IOC2. You can enable session affinity for the IPsec tunnel session on the IOC FPCs. To enable IPsec VPN affinity, you must also enable the session cache on IOCs by using the `set chassis fpc` *`fpc-slot`* `np-cache` command.

To enable VPN session affinity:

1. In configuration mode, use the `set` command to enable VPN session affinity.

   ```
   [edit]
   user@host# set security flow load-distribution session-affinity ipsec
   ```

2. Check your changes to the configuration before committing.

   ```
   [edit]
   user@host# commit check
   ```

3. Commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

After enabling VPN session affinity, use the `show security flow session` command to display session information about clear-text sessions.

```
user@host> show security flow session
Flow Sessions on FPC3 PIC0:

Session ID: 60000001, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/6352 --> 203.0.113.6/7927;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 60000002, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 60000003, Policy name: self-traffic-policy/1, Timeout: 56, Valid
  In: 203.0.113.6/500 --> 203.0.113.11/500;udp, If: .local..0, Pkts: 105425, Bytes: 12031144
  Out: 203.0.113.11/500 --> 203.0.113.6/500;udp, If: ge-0/0/2.0, Pkts: 106503, Bytes: 12110680
```

```
Session ID: 60017387, Policy name: default-policy-00/2, Timeout: 1796, Valid
  In: 192.0.2.155/53053 --> 198.51.100.156/23;tcp, If: ge-0/0/1.0, Pkts: 10, Bytes: 610
  Out: 198.51.100.156/23 --> 192.0.2.155/53053;tcp, If: st0.0, Pkts: 9, Bytes: 602
Total sessions: 4

Flow Sessions on FPC6 PIC0:

Session ID: 120000001, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0

Session ID: 120000002, Policy name: N/A, Timeout: N/A, Valid
  In: 203.0.113.11/0 --> 203.0.113.6/0;esp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0
Total sessions: 2
```

After VPN session affinity is enabled, the clear-text session is always located on FPC 3, PIC 0.

SEE ALSO

Understanding Session Cache

Express Path Overview

Accelerating the IPsec VPN Traffic Performance

You can accelerate IPsec VPN performance by configuring the performance acceleration parameter. By default, VPN performance acceleration is disabled on SRX Series Firewalls. Enabling the VPN performance acceleration can improve the VPN throughput with VPN session affinity enabled. This feature is only supported on SRX5400, SRX5600, and SRX5800 devices.

This topic describes how to use the CLI to enable VPN performance acceleration.

To enable performance acceleration, you must ensure that cleartext sessions and IPsec tunnel sessions are established on the same Services Processing Unit (SPU). Starting with Junos OS Release 17.4R1, IPsec VPN performance is optimized when the VPN session affinity and performance acceleration features are enabled. For more information on enabling session affinity, see "Understanding VPN Session Affinity" on page 1374.

To enable IPsec VPN performance acceleration:

1. Enable VPN session affinity.

```
[edit]
user@host# set security flow load-distribution session-affinity ipsec
```

2. Enable IPsec performance acceleration.

```
[edit]
user@host# set security flow ipsec-performance-acceleration
```

3. Check your changes to the configuration before committing.

```
[edit]
user@host# commit check
```

4. Commit the configuration.

```
[edit]
user@host# commit
```

After enabling VPN performance acceleration, use the `show security flow status` command to display flow status.

```
Flow forwarding mode:
   Inet forwarding mode: flow based
   Inet6 forwarding mode: drop
   MPLS forwarding mode: drop
   ISO forwarding mode: drop
 Flow trace status
   Flow tracing status: off
 Flow session distribution
   Distribution mode: Hash-based
       Flow packet ordering
   Ordering mode: Hardware
 Flow ipsec performance acceleration: on
```

SEE ALSO

- ipsec-performance-acceleration (Security Flow)
- show security flow status

IPsec Distribution Profile

Starting with Junos OS Release 19.2R1, you can configure one or more IPsec distribution profiles for IPsec security associations (SAs). Tunnels are distributed evenly across all resources (SPCs) specified in the configured distribution profile. It is supported in SPC3 only and mixed-mode (SPC3 + SPC2), it is not supported on SPC1 and SPC2 systems. With the IPsec distribution profile, use the `set security ipsec vpn` *`vpn-name`* `distribution-profile` *`distribution-profile-name`* command to associate tunnels to a specified:

- Slot
- PIC

Alternatively, you can use the default IPsec distribution profiles:

- `default-spc2-profile` —Use this predefined default profile to associate IPsec tunnels to all available SPC2 cards.
- `default-spc3-profile` —Use this predefined default profile to associate IPsec tunnels to all available SPC3 cards.

You can now assign a profile to a specific VPN object, where all associated tunnels will be distributed based on this profile. If no profile is assigned to the VPN object, the SRX Series Firewall automatically distributes these tunnels evenly across all resources.

You can associate a VPN object with either a user-defined profile or a predefined (default) profile.

Starting in Junos OS Release 20.2R2, the invalid thread IDs configured to the distribution profile are ignored with no commit-check error message. The IPsec tunnel gets anchored as per the configured distribution profile ignoring invalid thread IDs if any for that profile.

In the following example, all tunnels associated with profile ABC will be distributed on FPC 0, PIC 0.

```
userhost# show security {
    distribution-profile ABC {
        fpc 0 {
            pic 0;
        }
    }
}
```

Understanding the Loopback Interface for a High Availability VPN

In an IPsec VPN tunnel configuration, an external interface must be specified to communicate with the peer IKE gateway. Specifying a loopback interface for the external interface of a VPN is a good practice when there are multiple physical interfaces that can be used to reach a peer gateway. Anchoring a VPN tunnel on the loopback interface removes the dependency on a physical interface for successful routing.

Using a loopback interface for VPN tunnels is supported on standalone SRX Series Firewalls as well as on SRX Series Firewalls in chassis clusters. In a chassis cluster active-passive deployment, you can create a logical loopback interface and make it a member of a redundancy group so that it can be used to anchor VPN tunnels. The loopback interface can be configured in any redundancy group and is assigned as the external interface for the IKE gateway. VPN packets are processed on the node where the redundancy group is active.

On SRX5400, SRX5600, and SRX5800 devices -

- For SPC2 based devices running kmd process, if the loopback interface is used as the IKE gateway external interface, configure the interface binding in a redundancy group other than RG0.
- For SPC3 or SPC3+SPC2 based devices running iked process, loopback interface binding to a redundancy group is not required.

In a chassis cluster setup, the node on which the external interface is active selects an SPU to anchor the VPN tunnel. IKE and IPsec packets are processed on that SPU. Thus an active external interface determines the anchor SPU.

You can use the `show chassis cluster interfaces` command to view information on the redundant pseudointerface.

SEE ALSO

> *show chassis cluster interfaces*

Release History Table

Release	Description
12.3X48-D50	Starting with Junos OS Release 12.3X48-D50, Junos OS Release 15.1X49-D90, and Junos OS Release 17.3R1, if VPN session affinity is enabled on SRX5400, SRX5600, and SRX5800 devices, the tunnel overhead is calculated according to the negotiated encryption and authentication algorithms on the anchor Services Processing Unit (SPU).
17.4R1	Starting with Junos OS Release 17.4R1, IPsec VPN performance is optimized when the VPN session affinity and performance acceleration features are enabled.
Junos OS Release 20.2R	Starting in Junos OS Release 20.2R2, the invalid thread IDs configured to the distribution profile are ignored with no commit-check error message. The IPsec tunnel gets anchored as per the configured distribution profile ignoring invalid thread IDs if any for that profile.

RELATED DOCUMENTATION

VPN Support for Inserting Services Processing Cards | 175

IPsec VPN Configuration Overview | 190

PowerMode IPsec

IN THIS SECTION

Improving IPsec Performance with PowerMode IPsec

IN THIS SECTION

PowerMode IPsec (PMI) is a mode of operation that provides IPsec performance improvements using Vector Packet Processing and Intel Advanced Encryption Standard New Instructions (AES-NI). PMI utilizes a small software block inside the Packet Forwarding Engine that bypasses flow processing and utilizes the AES-NI instruction set for optimized performance of IPsec processing that gets activated when PMI is enabled.

PMI Processing

You can enable or disable PMI processing:

- Enable PMI processing by using the `set security flow power-mode-ipsec` configuration mode command.
- Disable PMI processing by using the `delete security flow power-mode-ipsec` configuration mode command. Executing this command deletes the statement from the configuration.

For SRX4100, SRX4200 devices running Junos OS Release 18.4R1, SRX4600 Series Firewalls running Junos OS Release 20.4R1, and vSRX Virtual Firewall running Junos OS Release 18.3R1 after you enable or disable the PMI, you must reboot the device for the configuration to take effect. However, for SRX5000 line and vSRX Virtual Firewall instances running Junos OS Release 19.2R1, reboot is not required.

PMI Statistics

You can verify the PMI statistics by using the `show security flow pmi statistics` operational mode command.

You can verify the PMI and fat tunnel status by using the `show security flow status` operational mode command.

Advanced Encryption Standard New Instructions (AES-NI) and Inline Field-Programmable Gate Array (FPGA)

Starting in Junos OS Release 20.4R1, you can enhance PMI performance by using AES-NI. AES-NI in PMI mode helps in balancing the load in SPUs and supports the symmetric fat tunnel in SPC3 cards. This results in accelerated traffic-handling performance and higher throughput for IPsec VPN. PMI uses AES-NI for encryption and FPGA for decryption of cryptographic operation.

To enable PMI processing with AES-NI, include the `power-mode-ipsec` statement at the `[edit security flow]` hierarchy level.

To enable or disable inline FPGA, include the `inline-fpga-crypto (disabled | enabled)` statement at the `[edit security forwarding-process application-services]` hierarchy level.

Supported and Non-Supported Features for PMI

A tunnel session can either be PMI or non-PMI.

If a session is configured with any non-supported features listed in Table 120 on page 1384 and Table 121 on page 1386, the session is marked as non-PMI and the tunnel goes into non-PMI mode. Once the tunnel goes into the non-PMI mode, the tunnel does not return to the PMI mode.

Table 120 on page 1384 summarizes the supported and non-supported PMI features on SRX Series Firewalls.

Table 120: Summary of Supported and Non-supported Features in PMI (SRX Series Firewalls)

Supported Features in PMI	Non-Supported Features in PMI
Internet Key Exchange (IKE) functionality	IPsec-in-IPsec tunnels
AutoVPN with traffic selectors	Layer 4 - 7 applications: application firewall and AppSecure
High availability	GPRS tunneling protocol (GTP) and Stream Control Transmission Protocol (SCTP) firewalls

Table 120: Summary of Supported and Non-supported Features in PMI (SRX Series Firewalls) *(Continued)*

Supported Features in PMI	Non-Supported Features in PMI
IPv6	Host traffic
Stateful firewall	Multicast
st0 interface	Nested tunnels
Traffic selectors	Screen options
NAT-T	DES-CBC encryption algorithm
GTP-U scenario with TEID distribution and asymmetric fat tunnel solution	3DES-CBC encryption algorithm
Quality of Service (QoS)	Application Layer Gateway (ALG)
First path and fast path processing for fragment handling and unified encryption.	
NAT	
AES-GCM-128 and AES-GCM-256 encryption algorithm. We recommend you to use AES-GCM encryption algorithm for optimal performance.	
AES-CBC-128, AES-CBC-192, and AES-CBC-256 with SHA1 encryption algorithm	
AES-CBC-128, AES-CBC-192, and AES-CBC-256 with SHA2 encryption algorithm	
NULL encryption algorithm	

Table 120: Summary of Supported and Non-supported Features in PMI (SRX Series Firewalls) *(Continued)*

Supported Features in PMI	Non-Supported Features in PMI

Table 121 on page 1386 summarizes the supported and non-supported PMI features on MX-SPC3 services card.

MX-SPC3 services card does not support np-cache and IPsec session-affinity.

Table 121: Summary of Supported and Non-supported Features in PMI (MX-SPC3 Services Card)

Supported Features in PMI	Non-Supported Features in PMI
Internet Key Exchange (IKE) functionality	Layer 4 - 7 applications: application firewall, AppSecure, and ALGs
AutoVPN with traffic selectors, ADVPN	Multicast
High availability	Nested tunnels
IPv6	Screen options
Stateful firewall	Application Layer Gateway (ALG)
st0 interface	
Traffic selectors	
Dead Peer Detection (DPD)	
Anti-Replay check	
NAT	
Post/Pre-Fragment	

Table 121: Summary of Supported and Non-supported Features in PMI (MX-SPC3 Services Card) *(Continued)*

Supported Features in PMI	Non-Supported Features in PMI
incoming clear-text fragments and ESP fragment	
AES-GCM-128 and AES-GCM-256 encryption algorithm. We recommend you to use AES-GCM encryption algorithm for optimal performance.	
AES-CBC-128, AES-CBC-192, and AES-CBC-256 with SHA1 encryption algorithm	
AES-CBC-128, AES-CBC-192, and AES-CBC-256 with SHA2 encryption algorithm	
NULL encryption algorithm	

Note the following usage considerations with PMI:

- Antireplay window size
  - Antireplay window size is 64 packets by default. If you configure fat-tunnel, then it is recommended to increase the Antireplay window size to greater than or equal to 512 packets.
- Class of Service (CoS)
  - Starting in Junos OS Release 19.1R1, Class of Service(CoS) supports configuration of behavior aggregate (BA) classifier, multifield (MF) classifier, and rewrite-rule functions in PMI on SRX5K-SPC3 Services Processing Card (SPC) cards.
  - If you enable PMI for a flow session, then the CoS is performed based on a per-flow basis. This means, the first packet of a new flow caches the CoS information in the flow session. Then the subsequent packets of the flow reuse the CoS information cached in the session.
- Encryption algorithm
  - Junos OS Release 19.3R1 supports options aes-128-cbc, aes-192-cbc, and aes-256-cbc on SRX4100, SRX4200, and vSRX Virtual Firewall in PMI mode to improve IPsec performance, along with the existing support in normal mode.
- GTP-U
  - Starting in Junos OS Release 19.2R1, PMI supports GTP-U scenario with TEID distribution and asymmetric fat tunnel solution.

  - Starting in Junos OS Release 19.3R1, GTP-U scenario with TEID distribution and asymmetric fat tunnel solution and Software Receive Side Scaling feature on vSRX Virtual Firewall and vSRX Virtual Firewall.

- LAG and redundant (reth) interfaces
  - PMI is supported on link aggregation group (LAG) and redundant Ethernet (reth) interfaces.

- PMI fragmentation check
  - PMI does a pre-fragmentation and post-fragmentation check. If the PMI detects pre-fragmentation and post-fragmentation packets, packets are not allowed through the PMI mode. The packets will return to non-PMI mode.
  - Any fragments received on an interface does not go through PMI.

- PMI for NAT-T
  - PMI for NAT-T is supported only on SRX5400, SRX5600, SRX5800 line equipped with SRX5K-SPC3 Services Processing Card (SPC), or with vSRX Virtual Firewall.

- PMI support (vSRX)
  - Starting in Junos OS Release 19.4R1, vSRX Virtual Firewall instances support:
    - Per-flow CoS functions for GTP-U traffic in PMI mode.
    - CoS features in PMI mode. The following CoS features are supported in PMI mode:
      - Classifier
      - Rewrite-rule functions
      - Queuing
      - Shaping
      - Scheduling

Benefits of PMI

- Enhances the performance of IPsec.

Configuring Security Flow PMI

The below section describes you how to configure security flow PMI.

To configure security flow PMI, you must enable session cache on IOCs and session affinity:

1. Enable the session cache on IOCs (IOC2 and IOC3)

   ```
   user@host# set chassis fpc <fpc-slot> np-cache
   ```

2. Enable VPN session affinity

   ```
   user@host# set security flow load-distribution session-affinity ipsec
   ```

3. Create security flow in PMI.

   ```
   user@host#set security flow power-mode-ipsec
   ```

4. Confirm your configuration by entering the `show security` command.

   ```
   user@host# show security
   flow {
       power-mode-ipsec;
   }
   ```

Understanding Symmetric Fat IPsec Tunnel

Starting from Junos OS 19.4R1, on SRX5400, SRX5600, and SRX5800 line with SRX5K-SPC3 service card, and vSRX Virtual Firewall instances, fat tunnel technology is introduced to improve the a IPsec tunnel throughput value up to 10 times of current value.

Starting in Junos OS Release 21.1R1, you can configure fat IPsec tunnel on MX-SPC3 services card.

A new CLI command is introduced to enable the fat IPsec tunnel. The fat IPsec tunnel feature is disabled by default. The new CLI command introduced is `fat-core` in the `set security distribution-profile` hierarchy. When you enable the fat-core, the below configuration is displayed:

```
security {
    distribution-profile {
        fat-core;
    }
}
```

Before configuring the fat IPsec tunnel, make sure the following are configured.

- For fast path forwarding, configure the IOC cache for the session information using the `set chassis fpc FPC slot np-cache` command.
- To enable session affinity, use the `set security flow load-distribution session-affinity ipsec` command.
- To enable Power mode, use the `set security flow power-mode-ipsec` command.

SEE ALSO

IPsec VPN Overview | 165
flow (Security Flow)
PMI Flow Based CoS functions for GTP-U
show security flow pmi statistics
inline-fpga-crypto | 1540
distribution-profile | 1490

Example: Configuring Behavior Aggregate Classifier in PMI

IN THIS SECTION

This example shows how to configure behavior aggregate(BA) classifiers for a SRX Series Firewall to determine forwarding treatment of packets in PMI.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall.
- Junos OS Release 19.1R1 and later releases.

Before you begin:

- Determine the forwarding class and PLP that are assigned by default to each well-known DSCP that you want to configure for the behavior aggregate classifier.

Overview

Configure behavior aggregate classifiers to classify the packets that contain valid DSCPs to appropriate queues. Once configured, you apply the behavior aggregate classifier to the correct interfaces. You override the default IP precedence classifier by defining a classifier and applying it to a logical interface. To define new classifiers for all code point types, include the `classifiers` statement at the `[edit class-of-service]` hierarchy level.

In this example, set the DSCP behavior aggregate classifier to `ba-classifier` as the default DSCP map. Set a best-effort forwarding class as `be-class`, an expedited forwarding class as `ef-class`, an assured forwarding class as `af-class`, and a network control forwarding class as `nc-class`. Finally, apply the behavior aggregate classifier to the interface ge-0/0/0.

Table 2 shows how the behavior aggregate classifier assigns loss priorities, to incoming packets in the four forwarding classes.

Table 122: Sample ba-classifier Loss Priority Assignments

mf-classifier Forwarding Class	For CoS Traffic Type	ba-classifier Assignments
`be-class`	Best-effort traffic	High-priority code point: 000001
`ef-class`	Expedited forwarding traffic	High-priority code point: 101111
`af-class`	Assured forwarding traffic	High-priority code point: 001100
`nc-class`	Network control traffic	High-priority code point: 110001

Configuration

IN THIS SECTION

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from the configuration mode.

```
set class-of-service classifiers dscp ba-classifier import default
set class-of-service classifiers dscp ba-classifier forwarding-class be-class loss-priority high
code-points 000001
set class-of-service classifiers dscp ba-classifier forwarding-class ef-class loss-priority high
code-points 101111
set class-of-service classifiers dscp ba-classifier forwarding-class af-class loss-priority high
code-points 001100
set class-of-service classifiers dscp ba-classifier forwarding-class nc-class loss-priority high
code-points 110001
set class-of-service interfaces ge-0/0/0 unit 0 classifiers dscp ba-classifier
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure Behavior Aggregate Classifiers for a device in PMI:

1. Configure the class of service.

   ```
   [edit]
   user@host# edit class-of-service
   ```

2. Configure behavior aggregate classifiers for Differentiated Services (DiffServ) CoS.

   ```
   [edit class-of-service]
   user@host# edit classifiers dscp ba-classifier
   user@host# set import default
   ```

3. Configure a best-effort forwarding class classifier.

   ```
   [edit class-of-service classifiers dscp ba-classifier]
   user@host# set forwarding-class be-class loss-priority high code-points 000001
   ```

4. Configure an expedited forwarding class classifier.

   ```
   [edit class-of-service classifiers dscp ba-classifier]
   user@host# set forwarding-class ef-class loss-priority high code-points 101111
   ```

5. Configure an assured forwarding class classifier.

   ```
   [edit class-of-service classifiers dscp ba-classifier]
   user@host# set forwarding-class af-class loss-priority high code-points 001100
   ```

6. Configure a network control forwarding class classifier.

   ```
   [edit class-of-service classifiers dscp ba-classifier]
   user@host# set forwarding-class nc-class loss-priority high code-points 110001
   ```

7. Apply the behavior aggregate classifier to an interface.

   ```
   [edit]
   user@host# set class-of-service interfaces ge-0/0/0 unit 0 classifiers dscp ba-classifier
   ```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service
classifiers {
    dscp ba-classifier {
        import default;
        forwarding-class be-class {
            loss-priority high code-points 000001;
        }
        forwarding-class ef-class {
            loss-priority high code-points 101111;
        }
        forwarding-class af-class {
            loss-priority high code-points 001100;
        }
        forwarding-class nc-class {
            loss-priority high code-points 110001;
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            classifiers {
                dscp ba-classifier;
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the Classifier is applied to the Interfaces

Purpose

Make sure that the classifier is applied to the correct interfaces.

Action

From the operational mode, enter the `show class-of-service interface ge-0/0/0` command.

```
user@host> show class-of-service interface ge-0/0/0
Physical interface: ge-0/0/0, Index: 144
 Queues supported: 8, Queues in use: 4
Scheduled map: <default>, Index:2
Congestion-notification: Disabled

LOgical interface: ge-1/0/3, Index: 333
Object      Name              Type    Index
Classifier   v4-ba-classifier  dscp    10755
```

Meaning

The interfaces are configured as expected.

Example: Configuring Behavior Aggregate Classifier in PMI for vSRX Virtual Firewall instances

IN THIS SECTION

This example shows how to configure behavior aggregate (BA) classifiers for a vSRX Virtual Firewall instance to determine forwarding treatment of packets in PMI.

Requirements

This example uses the following hardware and software components:

- A vSRX Virtual Firewall instance.
- Junos OS Release 19.4R1 and later releases.

Before you begin:

- Determine the forwarding class and Packet loss priorities(PLP) that are assigned by default to each well-known DSCP that you want to configure for the behavior aggregate classifier.

Overview

Configure behavior aggregate classifiers to classify the packets that contain valid DSCPs to appropriate queues. Once configured, you apply the behavior aggregate classifier to the correct interfaces. You override the default IP precedence classifier by defining a classifier and applying it to a logical interface. To define new classifiers for all code point types, include the `classifiers` statement at the `[edit class-of-service]` hierarchy level.

In this example, set the DSCP behavior aggregate classifier to `ba-classifier` as the default DSCP map. Set a best-effort forwarding class as `be-class`, an expedited forwarding class as `ef-class`, an assured forwarding class as `af-class`, and a network control forwarding class as `nc-class`. Finally, apply the behavior aggregate classifier to the interface ge-0/0/0.

Table 2 shows how the behavior aggregate classifier assigns loss priorities, to incoming packets in the four forwarding classes.

Table 123: Sample ba-classifier Loss Priority Assignments

mf-classifier Forwarding Class	For CoS Traffic Type	ba-classifier Assignments
`be-class`	Best-effort traffic	High-priority code point: 000001
`ef-class`	Expedited forwarding traffic	High-priority code point: 101111
`af-class`	Assured forwarding traffic	High-priority code point: 001100
`nc-class`	Network control traffic	High-priority code point: 110001

Configuration

IN THIS SECTION

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from the configuration mode.

```
set class-of-service classifiers dscp ba-classifier forwarding-class be loss-priority low code-
points be
set class-of-service classifiers dscp ba-classifier forwarding-class ef loss-priority low code-
points ef
set class-of-service classifiers dscp ba-classifier forwarding-class ef loss-priority high code-
points af41
set class-of-service classifiers dscp ba-classifier forwarding-class ef loss-priority high code-
points af11
set class-of-service classifiers dscp ba-classifier forwarding-class ef loss-priority high code-
```

```
points af31
set class-of-service classifiers dscp ba-classifier forwarding-class low_delay loss-priority low
code-points af21
set class-of-service classifiers dscp ba-classifier forwarding-class low_loss loss-priority low
code-points cs6
set class-of-service drop-profiles drop_profile fill-level 20 drop-probability 50
set class-of-service drop-profiles drop_profile fill-level 50 drop-probability 100
set class-of-service forwarding-classes queue 0 be
set class-of-service forwarding-classes queue 1 ef
set class-of-service forwarding-classes queue 2 low_delay
set class-of-service forwarding-classes queue 3 low_loss
set class-of-service interfaces ge-0/0/1 unit 0 classifiers dscp ba-classifier
set class-of-service interfaces ge-0/0/3 unit 0 scheduler-map SCHEDULER-MAP
set class-of-service interfaces ge-0/0/3 unit 0 shaping-rate 2k
set class-of-service scheduler-maps SCHEDULER-MAP forwarding-class ef scheduler voice
set class-of-service schedulers voice buffer-size temporal 5k
set class-of-service schedulers voice drop-profile-map loss-priority any protocol any drop-
profile drop_profile
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure Behavior Aggregate Classifiers for a device in PMI:

1. Configure the class of service.

```
[edit]
user@host# edit class-of-service
```

2. Configure behavior aggregate classifiers for Differentiated Services (DiffServ) CoS.

```
[edit class-of-service]
user@host# edit classifiers dscp ba-classifier
```

3. Configure a best-effort forwarding class classifier.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class be loss-priority low code-points be
```

4. Configure an expedited forwarding class classifier.

```
[edit class-of-service classifiers dscp ba-classifier]
user@host# set forwarding-class ef-class loss-priority low code-points ef
user@host# set forwarding-class ef-class loss-priority high code-points af41
user@host# set forwarding-class ef-class loss-priority high code-points af11
user@host# set forwarding-class ef-class loss-priority high code-points af31
user@host# set forwarding-class low_delay loss-priority low code-points af21
user@host# set forwarding-class low_loss loss-priority low code-points cs6
```

5. Configure drop profiles.

```
[edit class-of-service drop-profiles]
user@host# set drop_profile fill-level 20 drop-probability 50
user@host# set drop_profile fill-level 50 drop-probability 100
```

6. Configure the forwarding classes queues.

```
[edit class-of-service forwarding-classes ]
user@host# set queue 0 be
user@host# set queue 1 ef
user@host# set queue 2 low_delay
user@host# set 3 low_loss
```

7. Apply the classifier to the interfaces.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/1 unit 0 classifiers dscp ba-classifier
user@host# set interfaces ge-0/0/3 unit 0 scheduler-map SCHEDULER-MAP
user@host# set interfaces ge-0/0/3 unit 0 shaping-rate 2k
```

8. Configure the schedulers.

```
[edit class-of-service]
user@host# set scheduler-maps SCHEDULER-MAP forwarding-class ef scheduler voice
user@host# set schedulers voice buffer-size temporal 5k
user@host# set schedulers voice drop-profile-map loss-priority any protocol any drop-profile
drop_profile
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service
classifiers {
    dscp ba-classifier {
        forwarding-class be {
            loss-priority low code-points be;
        }
        forwarding-class ef {
            loss-priority low code-points ef;
            loss-priority high code-points [ af41 af11 af31 ];
        }
        forwarding-class low_delay {
            loss-priority low code-points af21;
        }
        forwarding-class low_loss {
            loss-priority low code-points cs6;
        }
    }
}
drop-profiles {
    drop_profile {
        fill-level 20 drop-probability 50;
        fill-level 50 drop-probability 100;
    }
}
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 low_delay;
    queue 3 low_loss;
}
interfaces {
    ge-0/0/1 {
        unit 0 {
            classifiers {
                dscp ba-classifier;
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            scheduler-map SCHEDULER-MAP;
            shaping-rate 2k;
        }
    }
}
scheduler-maps {
    SCHEDULER-MAP {
        forwarding-class ef scheduler voice;
    }
}
schedulers {
    voice {
        buffer-size temporal 5k;
        drop-profile-map loss-priority any protocol any drop-profile drop_profile;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying the Classifier is applied to the Interfaces

Purpose

Verify that the classifier is configured properly and confirm that the forwarding classes are configured correctly.

Action

From the operational mode, enter the `show class-of-service forwarding-class` command.

```
user@host> show class-of-service forwarding-class
Forwarding class                       ID      Queue  Restricted queue  Fabric priority
Policing priority   SPU priority
  be                                    0       0          0             low
normal            low
  ef                                    1       1          1             low
normal            low
  low_delay                             2       2          2             low
normal            low
  low_loss                              3       3          3             low
normal            low
```

Meaning

The output shows the configured custom classifier settings.

Example: Configuring and Applying a Firewall Filter for a Multifield Classifier in PMI

IN THIS SECTION

This example shows how to configure a firewall filter to classify traffic to different forwarding class by using DSCP value and multifield (MF) classifier in PMI.

The classifier detects packets of interest to class of service (CoS) as they arrive on an interface. MF classifiers are used when a simple behavior aggregate (BA) classifier is insufficient to classify a packet, when peering routers do not have CoS bits marked, or the peering router's marking is untrusted.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall.
- Junos OS Release 19.1R1 and later releases.

Before you begin:

- Determine the forwarding class that are assigned by default to each well-known DSCP that you want to configure for the MF classifier. See "Improving IPsec Performance with PowerMode IPsec" on page 1383.

Overview

This example explain how to configure the firewall filter `mf-classifier`. To configure the MF classifier, create and name the assured forwarding traffic class, set the match condition, and then specify the destination address as 192.168.44.55. Create the forwarding class for assured forwarding DiffServ traffic as `af-class` and set the loss priority to low.

In this example, create and name the expedited forwarding traffic class and set the match condition for the expedited forwarding traffic class. Specify the destination address as 192.168.66.77. Create the forwarding class for expedited forwarding DiffServ traffic as `ef-class` and set the policer to `ef-policer`. Create and name the network-control traffic class and set the match condition.

In this example, create and name the forwarding class for the network control traffic class as `nc-class` and name the forwarding class for the best-effort traffic class as `be-class`. Finally, apply the multifield classifier firewall filter as an input and output filter on each customer-facing or host-facing that needs the filter. In this example, the interface for input filter is ge-0/0/2 and interface for output filter is ge-0/0/4.

Configuration

IN THIS SECTION

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from the configuration mode.

```
set firewall filter mf-classifier interface-specific
set firewall filter mf-classifier term assured-forwarding from destination-address 192.168.44.55
set firewall filter mf-classifier term assured-forwarding then forwarding-class af-class
set firewall filter mf-classifier term assured-forwarding then loss-priority low
set firewall filter mf-classifier term expedited-forwarding from destination-address
192.168.66.77
set firewall filter mf-classifier term expedited-forwarding then forwarding-class ef-class
set firewall filter mf-classifier term expedited-forwarding then policer ef-policer
set firewall filter mf-classifier term network-control from precedence net-control
set firewall filter mf-classifier term network-control then forwarding-class nc-class
set firewall filter mf-classifier term best-effort then forwarding-class be-class
set interfaces ge-0/0/2 unit 0 family inet filter input mf-classifier
set interfaces ge-0/0/4 unit 0 family inet filter output mf-classifier
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure a Firewall Filter for a Multifield Classifier for a device in PMI:

1. Create and name the multifield classifier filter.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# set interface-specific
```

2. Create and name the term for the assured forwarding traffic class.

```
[edit firewall filter mf-classifier]
user@host# edit term assured-forwarding
```

3. Specify the destination address for assured forwarding traffic.

```
[edit firewall filter mf-classifier term assured-forwarding]
user@host# set from destination-address 192.168.44.55
```

4. Create the forwarding class and set the loss priority for the assured forwarding traffic class.

```
[edit firewall filter mf-classifier term assured-forwarding]
user@host# set then forwarding-class af-class
user@host# set then loss-priority low
```

5. Create and name the term for the expedited forwarding traffic class.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# edit term expedited-forwarding
```

6. Specify the destination address for the expedited forwarding traffic.

```
[edit firewall filter mf-classifier term expedited-forwarding]
user@host# set from destination-address 192.168.66.77
```

7. Create the forwarding class and apply the policer for the expedited forwarding traffic class.

```
[edit firewall filter mf-classifier term expedited-forwarding]
user@host# set then forwarding-class ef-class
user@host# set then policer ef-policer
```

8. Create and name the term for the network control traffic class.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# edit term network-control
```

9. Create the match condition for the network control traffic class.

```
[edit firewall filter mf-classifier term network-control]
user@host# set from precedence net-control
```

10. Create and name the forwarding class for the network control traffic class.

```
[edit firewall filter mf-classifier term network-control]
user@host# set then forwarding-class nc-class
```

11. Create and name the term for the best-effort traffic class.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# edit term best-effort
```

12. Create and name the forwarding class for the best-effort traffic class.

```
[edit firewall filter mf-classifier term best-effort]
user@host# set then forwarding-class be-class
```

13. Apply the multifield classifier firewall filter as an input filter.

```
[edit]
user@host# set interfaces ge-0/0/2 unit 0 family inet filter input mf-classifier
```

14. Apply the multifield classifier firewall filter as an output filter.

```
[edit]
user@host# set interfaces ge-0/0/4 unit 0 family inet filter output mf-classifier
```

Results

From configuration mode, confirm your configuration by entering the `show firewall filter mf-classifier` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show firewall filter mf-classifier
interface-specific;
    term assured-forwarding {
    from {
        destination-address {
            192.168.44.55/32;
        }
    }
    then {
        loss-priority low;
        forwarding-class af-class;
    }
}
term expedited-forwarding {
    from {
        destination-address {
            192.168.66.77/32;
        }
    }
    then {
        policer ef-policer;
        forwarding-class ef-class;
    }
}
term network-control {
    from {
        precedence net-control;
    }
    then forwarding-class nc-class;
}
term best-effort {
    then forwarding-class be-class;
}
```

From configuration mode, confirm your configuration by entering the `show interfaces` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show show interfaces
ge-0/0/2 {
    unit 0 {
        family inet {
            filter {
                input mf-classifier;
            }
        }
    }
}
ge-0/0/4 {
    unit 0 {
        family inet {
            filter {
                output mf-classifier;
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying a Firewall Filter for a Multifield Classifier Configuration

Purpose

Verify that a firewall filter for a multifield classifier is configured properly on a device and confirm that the forwarding classes are configured correctly.

Action

From configuration mode, enter the `show class-of-service forwarding-class` command.

```
user@host> show class-of-service forwarding-class
Forwarding class                       ID      Queue  Restricted queue  Fabric priority
Policing priority   SPU priority
  BE-data                               0       0          0             low
normal            low
  Premium-data                          1       1          1             low
normal            low
  Voice                                 2       2          2             low
normal            low
  NC                                    3       3          3             low
normal            low
```

Meaning

The output shows the configured custom classifier settings.

Example: Configuring and Applying Rewrite Rules on a Security Device in PMI

IN THIS SECTION

This example shows how to configure and apply rewrite rules for a device in PMI.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall.
- Junos OS Release 19.1R1 and later releases.

Before you begin:

- Create and configure the forwarding classes. See "Improving IPsec Performance with PowerMode IPsec" on page 1383.

Overview

This example explains how to configure rewrite rules to replace CoS values on packets received from the customer or host with the values expected by other SRX Series Firewalls. You do not have to configure rewrite rules if the received packets already contain valid CoS values. Rewrite rules apply the forwarding class information and packet loss priority used internally by the device to establish the CoS value on outbound packets. After you configure the rewrite rules, apply them to the correct interfaces.

In this example, configure the rewrite rule for DiffServ CoS as `rewrite-dscps`. Specify the best-effort forwarding class as `be-class`, expedited forwarding class as `ef-class`, an assured forwarding class as `af-class`, and a network control class as `nc-class`. Finally, apply the rewrite rule to the ge-0/0/0 interface.

Configuration

IN THIS SECTION

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from the configuration mode.

```
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class be-class loss-priority
low code-point 000000
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class be-class loss-priority
high code-point 000001
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class ef-class loss-priority
low code-point 101110
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class ef-class loss-priority
high code-point 101111
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class af-class loss-priority
low code-point 001010
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class af-class loss-priority
high code-point 001100
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class nc-class loss-priority
low code-point 110000
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class nc-class loss-priority
high code-point 110001
set class-of-service interfaces ge-0/0/0 unit 0 rewrite-rules dscp rewrite-dscps
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure and apply Rewrite Rules for a device in PMI:

1. Configure rewrite rules for DiffServ CoS.

   ```
   [edit]
   user@host# edit class-of-service
   user@host# edit rewrite-rules dscp rewrite-dscps
   ```

2. Configure best-effort forwarding class rewrite rules.

   ```
   [edit class-of-service rewrite-rules dscp rewrite-dscps]
   user@host# set forwarding-class be-class loss-priority low code-point 000000
   user@host# set forwarding-class be-class loss-priority high code-point 000001
   ```

3. Configure expedited forwarding class rewrite rules.

   ```
   [edit class-of-service rewrite-rules dscp rewrite-dscps]
   user@host# set forwarding-class ef-class loss-priority low code-point 101110
   user@host# set forwarding-class ef-class loss-priority high code-point 101111
   ```

4. Configure an assured forwarding class rewrite rules.

   ```
   [edit class-of-service rewrite-rules dscp rewrite-dscps]
   user@host# set forwarding-class af-class loss-priority low code-point 001010
   user@host# set forwarding-class af-class loss-priority high code-point 001100
   ```

5. Configure a network control class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class nc-class loss-priority low code-point 110000
user@host# set forwarding-class nc-class loss-priority high code-point 110001
```

6. Apply rewrite rules to an interface.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/0  unit 0 rewrite-rules dscp rewrite-dscps
```

Results

From configuration mode, confirm your configuration by entering the `show class-of-service` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service
interfaces {
    ge-0/0/0 {
        unit 0 {
            rewrite-rules {
                dscp rewrite-dscps;
            }
        }
    }
}
rewrite-rules {
    dscp rewrite-dscps {
        forwarding-class be-class {
            loss-priority low code-point 000000;
            loss-priority high code-point 000001;
        }
        forwarding-class ef-class {
            loss-priority low code-point 101110;
            loss-priority high code-point 101111;
        }
        forwarding-class af-class {
            loss-priority low code-point 001010;
            loss-priority high code-point 001100;
        }
        forwarding-class nc-class {
            loss-priority low code-point 110000;
            loss-priority high code-point 110001;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying Rewrite Rules Configuration

Purpose

Verify that rewrite rules are configured properly.

Action

From the operational mode, enter the `show class-of-service` command.

```
user@host> show class-of-service
Physical interface: ge-0/0/0, Index: 130
 Maximum usable queues: 8, Queues in use: 4
Scheduled map: <default>, Index:2
Congestion-notification: Disabled

LOgical interface: ge0/0/0, Index: 71
Object      Name                   Type    Index
Classifier   ipprec-compatibility  ip    13
```

Meaning

Rewrite rules are configured on ge-0/0/0 interface as expected.

Configure IPsec ESP Authentication-only Mode in PMI

The PMI introduced a new data path for achieving a high IPsec throughput performance. Starting in Junos OS Release 19.4R1, on SRX5000 line with SRX5K-SPC3 card, you can use Encapsulating Security Payload (ESP) authentication-only mode in PMI mode, which provides authentication, integrity checking, and replay protection without encrypting the data packets.

Starting in Junos OS release 22.1R3, we support the PMI express path processing for passthrough ESP traffic on the SRX Series Firewalls.

Before you begin:

- Make sure that the session is PMI capable. See "VPN Session Affinity " on page 1374.

To configure ESP authentication-only mode:

1. Configure IPsec proposal and policy.

```
user@host# set security ipsec proposal IPSEC_PROP protocol esp
user@host# set security ipsec proposal IPSEC_PROP authentication-algorithm hmac-sha-256-128
user@host# set security ipsec policy IPSEC_POL proposals IPSEC_PROP
```

2. Confirm your configuration by entering the `show security ipsec` command.

```
user@host# show security ipsec
proposal IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
}
policy IPSEC_POL {
    proposals IPSEC_PROP;
}
```

   If you are done configuring the device, enter `commit` from configuration mode.

SEE ALSO

proposal (Security IPsec) | 1602

17

CHAPTER

Troubleshooting

Troubleshoot a Flapping VPN Tunnel

IN THIS SECTION

Problem

Description

Site-to-site VPN tunnel or remote IPsec VPN tunnel flapping (that is, going up and down in quick succession).

Diagnosis

1. Does the issue affect only one VPN?

    - Yes: Check the system logs and proceed to Step 2. Use the `show log messages` command to view the logs. You must enable information-level logging for messages to be reported correctly.

      ```
      user@host # set system syslog file messages any info
      ```

      Here are examples of system logs reporting a flapping VPN tunnel:

      VPN up/down events:

      ```
      Jul 9 21:07:58 kmd[1496]: KMD_VPN_DOWN_ALARM_USER: VPN to_hub from 3.3.3.2 is down. Local-ip: 4.4.4.4, gateway name: to_hub, vpn name: to_hub, tunnel-id: 131073, local tunnel-if: st0.0, remote tunnel-ip: 70.70.70.1, Local IKE-ID: 4.4.4.4, Remote IKE-ID: 3.3.3.2, XAUTH username: Not-Applicable, VR id: 4
      Jul 9 21:08:10 kmd[1496]: KMD_VPN_UP_ALARM_USER: VPN to_hub from 3.3.3.2 is up. Local-ip: 4.4.4.4, gateway name: to_hub, vpn name: to_hub, tunnel-id: 131073, local tunnel-if: st0.0, remote tunnel-ip: 70.70.70.1, Local IKE-ID: 4.4.4.4, Remote IKE-ID: 3.3.3.2, XAUTH username: Not-Applicable, VR id: 4
      ```

```
Jul 9 21:09:58 kmd[1496]: KMD_VPN_DOWN_ALARM_USER: VPN to_hub from 3.3.3.2 is down. Local-
ip: 4.4.4.4, gateway name: to_hub, vpn name: to_hub, tunnel-id: 131073, local tunnel-if:
st0.0, remote tunnel-ip: 70.70.70.1, Local IKE-ID: 4.4.4.4, Remote IKE-ID: 3.3.3.2, XAUTH
username: Not-Applicable, VR id: 4
Jul 9 21:10:10 kmd[1496]: KMD_VPN_UP_ALARM_USER: VPN to_hub from 3.3.3.2 is up. Local-ip:
4.4.4.4, gateway name: to_hub, vpn name: to_hub, tunnel-id: 131073, local tunnel-if:
st0.0, remote tunnel-ip: 70.70.70.1, Local IKE-ID: 4.4.4.4, Remote IKE-ID: 3.3.3.2, XAUTH
username: Not-Applicable, VR id: 4
```

Unstable VPN behavior (VPN constantly rebuilding):

```
Jul 9 20:43:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: inbound, SPI: 0xfd91b643, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
Jul 9 20:43:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: outbound, SPI: 0xbdec9669, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
Jul 9 20:44:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: inbound, SPI: 0x69b34ae4, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
Jul 9 20:44:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: outbound, SPI: 0x6f55d8ea, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
Jul 9 20:45:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: inbound, SPI: 0x6fa6b0b3, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
Jul 9 20:45:10 kmd[1496]: KMD_PM_SA_ESTABLISHED: Local gateway: 4.4.4.4, Remote gateway:
3.3.3.2, Local ID: ipv4_subnet(any:0,[0..7]=0.0.0.0/0), Remote ID: ipv4_subnet(any:0,
[0..7]=0.0.0.0/0), Direction: outbound, SPI: 0xa66ac906, AUX-SPI: 0, Mode: Tunnel, Type:
dynamic
```

- No: If the issue is on all configured VPNs, investigate the errors associated with the Internet connection, and on the SRX Series Firewall and switch interfaces. To check for errors on the SRX Series Firewall interface, run the `show interfaces extensive` command.

2. Verify that VPN Monitor is enabled for this VPN by using the `show configuration security ipsec vpn` *`vpn-name`* command.

Is VPN Monitor enabled?

- Yes: Proceed to Step 3.
- No: Proceed to Step 5.

3. Disable VPN Monitor and check the VPN.

   ```
   user@host# deactivate security ipsec vpn vpn-name vpn-monitor
   user@host# commit
   ```

   Is the VPN stable?

   - Yes: The instability is related to the VPN Monitor configuration. Proceed to Step 4.
   - No: Proceed to Step 5.

4. Is the remote VPN connection configured to block ICMP echo requests?

   - Yes: Reenable and reconfigure VPN Monitor to use the source interface and destination IP options. See KB10119.
   - No: Proceed to Step 5.

5. Is the remote device that is connected to the SRX Series Firewall a non-Juniper device?

   - Yes: Verify the *proxy-id* value on the SRX Series Firewall and the peer VPN device.
   - No: Proceed to Step 6.

6. Was the VPN stable for a period of time and then started going up and down?

   - Yes: Investigate for network or device changes or whether any new network equipment has been added to the environment.
   - No: Collect site-to-site logs from the VPN devices at both ends and open a case with your technical support representative. See Data Collection for Customer Support.

Troubleshoot a VPN That Is Up But Not Passing Traffic

IN THIS SECTION

Problem

Description

The VPN is up, but there is no passing traffic in one or both directions.

This topic helps troubleshoot the issues that could prevent traffic passing through an active VPN tunnel.

Environment

VPN

Solution

1. Check whether the VPN security association (SA) is active: **show security ipsec security-associations**

```
user@CORPORATE> show security ipsec security-associations
  total configured sa: 1
  ID     Gateway         Port  Algorithm       SPI      Life:sec/kb  Mon vsys
  <32785 2.2.2.2         1398  ESP:3des/sha1   29e26eba 28735/unlim   -   0
  >32785 2.2.2.2         1398  ESP:3des/sha1   6d4e790b 28735/unlim   -   0
```

If the VPN gateway is listed, the tunnel is established and is up. The output displays two lines for each VPN tunnel displaying the SPI information for each direction of traffic.

The MON field is used by VPN monitoring to show the status of the tunnel and has one of the following values:

- **-** (hyphen): The VPN tunnel is active, and the VPN monitor optional feature is not configured.
- **U** (up): The VPN tunnel is active, and the link (detected through the VPN monitor) is up.
- **D** (down): The VPN tunnel is active, and the link (detected through the VPN monitor) is down.
- **Yes**: The IPsec SA state is active or up. Proceed to Step "2" on page 1422.
- **No**: The IPsec SA state is down. See How to troubleshoot a VPN tunnel that is down or not active.

2. Check whether the VPN is using the loopback interface lo0 as the external interface: **show configuration security ike**

```
root> show configuration security ike
policy ike_pol {
   proposal-set compatible;
   pre-shared-key ascii-text "$9$tMwDuIESreWX7yr4aGDkqIEhcvWbs2";
}
gateway gate1 {
  ike-policy ike_pol;
  address 10.10.10.2;
  external-interface lo0.0;
}
```

- **Yes**: VPN is using the the loopback interface **lo0** as the external interface. Proceed to Step "3" on page 1422.
- **No**: VPN is not using the the loopback interface **lo0** as the external interface. Proceed to Step "4" on page 1422.

3. Check whether the egress interface (physical interface) and lo0 used as the VPN external interface are in the same security zone.

- **Yes**: Proceed to Step "4" on page 1422.
- **No**: Update the security zone assignments so that both the VPN external interface and the physical egress interface are in the same security zone. See Traffic Loss when IPSec VPN is terminated on loopback interface.

4. If your VPN is a route-based VPN, proceed to Step "5" on page 1423. Proceed to Step "8" on page 1424 if it is a policy-based VPN. See What is the difference between a policy-based VPN and a route-based VPN?

5. Check whether a route is assigned to the remote network through the st0 interface: **show route *remote network***

```
root@siteA > show route 192.168.20.10
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
192.168.2.0/24    *[ARI-TS/5] 00:00:53
                           > via st0.0  <----------
```

- **Yes**: Proceed to Step "6" on page 1423.
- **No**: Assign a route to the remote network through the st0 interface. See Route-based VPN is up, but not passing traffic. Is a route missing?.

> **NOTE**: If you are using a dynamic routing protocol, such as BGP or OSPF, then check the routing protocol.

6. Based on the route assigned to the remote network in Step "5" on page 1423, check whether the VPN is pointing to the correct st0 interface: **show security ike** and **show security ipsec**

   a. First, check the IKE gateway using the **show security ike** command.

```
root@siteA # show security ike
...
gateway gw-siteB {        <---------
     ike-policy ike-phase1-policy;
     address 2.2.2.2;
     external-interface ge-0/0/3.0;
}
```

   b. Check the IPsec VPN for that IKE gateway using the **show security ipsec** command and in the output verify if `bind-interface` is pointing to `st0` interface.

   In this example, the VPN `ike-vpn-siteB` is pointing to the `st0.0` interface.

```
root@siteA # show security ipsec
...
vpn ike-vpn-siteB {
    bind-interface st0.0;
      ike {
            gateway gw-siteB;      <---------
            proxy-identity {
                local 192.168.2.0/24;
                remote 192.168.1.0/24;
                service any;
              }
             ipsec-policy ipsec-phase2-policy;
           }
        establish-tunnels immediately;
       }
```

- **Yes**: Proceed to Step "7" on page 1424.

- **No**: VPN is not pointing to the correct st0 interface. Delete the current route, and add the route to the correct st0 interface. See Route-based VPN is up, but not passing traffic. Is a route missing?.

7. Check whether there is a security policy that allows traffic from the internal zone to the st0 security zone: **show security policies**

   - **Yes**: Proceed to Step "8" on page 1424.

   - **No**: Create the appropriate security policy and test the VPN again. See How to configure a policy for a route-based VPN.

8. Check whether there is a VPN tunnel security policy to allow traffic: **show security policies**

```
root@siteA# show security policies
...
from-zone trust to-zone untrust {
    policy vpn_egress {
        match {
            source-address local-net;
            destination-address remote-net;
            application any;
        }
        then {
            permit {
                tunnel {                      <----------
                    ipsec-vpn ike-vpn-siteC;  <----------
                }
            }
        }
    }
 }

from-zone untrust to-zone trust {
    policy vpn_ingress {
        match {
            source-address remote-net;
            destination-address local-net;
            application any;
        }
        then {
            permit {
                tunnel {                      <----------
                    ipsec-vpn ike-vpn-siteC;  <----------
                }
            }
        }
    }
 }
```

- **Yes**: Proceed to Step "9" on page 1425.
- **No**: Verify the policy-based VPN configuration. See Policy-Based site-to-site VPN .

9. Check whether the traffic is matching in the policies identified in step "7" on page 1424 or step "8" on page 1424: **show security flow session source prefix *source address* destination prefix *destination address***

   ```
   root@siteA> show security flow session source-prefix 192.168.2.0/24 destination-prefix
   192.168.1.0/24

   Session ID: 5801, Policy name: AtoB/2, Timeout: 1790, Valid
   In: 192.168.2.222/1 --> 192.168.1.13/23053;icmp, If: fe-0/0/2.0, Pkts: 59878, Bytes: 4602292
   Out: 192.168.1.13/23053 --> 192.168.2.222/1;icmp, If: st0.0, Pkts: 52505, Bytes: 4189289
   ```

   - **Yes**: Proceed to Step "10" on page 1426.
   - **No**: Verify the order of the security policies: **show security match policies**. See Understanding Security Policy Ordering.

     If the order is correct, see How to troubleshoot a security policy that is not passing data.

NOTE: If only the `pkts` counter in the out direction of the session is incrementing, then validate with the VPN peer that the traffic is being received.

This is to check the packet counters on the VPN peer with which this tunnel is formed to see whether the other end is receiving the packets.

10. Collect logs and flow trace options and open a case with the Juniper Networks support team:

    - See the IPsec VPN policy-based or route-based VPN sections in Data Collection Checklist - Logs/data to collect for troubleshooting.

    - For information regarding flow trace options, see How to use 'flow traceoptions' and the 'security datapath-debug'.

    - To open a JTAC case with the Juniper Networks support team, see Data Collection for Customer Support for the data you should collect to assist in troubleshooting before opening a JTAC case.

Troubleshoot a VPN Tunnel That is Down

Problem: IPsec VPN is not active and does not pass data.

1. What type of VPN tunnel are you having trouble with?

   - Site-to-site (LAN-to-LAN) VPN:

     Proceed to Step 2.

   - Remote Access IPsec VPN or Client-to-LAN VPN:

     For branch SRX Series, see KB17220.

     For high-end SRX Series, proceed to Step 2.

2. Is the SA (security association) for the VPN tunnel active?

   Run the `show security ipsec security-associations` command and locate the gateway address of the VPN. If the remote gateway is not displayed, then the VPN SA is not active. For more information about SA, see KB10090.

```
user@host> show security ipsec security-associations
   total configured sa: 2
   ID     Gateway         Port  Algorithm       SPI      Life:sec/kb  Mon vsys
   <32785 2.2.2.2         1398  ESP:3des/sha1   29e26eba 28735/unlim   -   0
   >32785 2.2.2.2         1398  ESP:3des/sha1   6d4e790b 28735/unlim   -   0
   total configured sa: 2
   ID     Gateway         Port  Algorithm       SPI      Life:sec/kb  Mon vsys
   <32786 3.3.3.3         500   ESP:3des/sha1   5c13215d 28782/unlim   U   0
   >32786 3.3.3.3         500   ESP:3des/sha1   18f67b48 28782/unlim   U   0
```

- If SA is not listed in the output, proceed to Step 3.
- If SA is listed (Phase 2 is up) and if traffic is not passing, see "Troubleshoot a VPN That Is Up But Not Passing Traffic" on page 1421.
- If SA oscillates between active and inactive states, see "Troubleshoot a Flapping VPN Tunnel" on page 1418.

3. Is the IKE Phase 1 up?

   Run the `show security ike security-associations` command. Verify that the remote address of the VPN is listed and that the value of the `State` field is UP.

```
user@host> show security ike security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  Mode
1       2.2.2.2         UP     744a594d957dd513  1e1307db82f58387  Main
2       3.3.3.3         UP     744a594d957dd513  1e1307db82f58387  Main
```

   - If the remote address is not listed or if the value of the `State` field is `DOWN`, analyze the IKE Phase 1 messages on the responder for a solution. See KB10101.
   - If the state is `UP`, analyze the IKE Phase 2 messages on the responder for a solution. See KB10101.

   If the issue is still not resolved, analyze Phase 1 or Phase 2 logs for the VPN tunnel on the initiating VPN device. If you can't find your solution in the logs on the initiating side, proceed to Step 4.

4. Collect logs, flow trace options, and IKE trace options, and then open a case with your technical support representative. For information about:

   - Collecting logs, see Data Collection for Customer Support.
   - Flow trace options, see KB16233.
   - IKE trace options, see KB19943.

How to Analyze IKE Phase 2 VPN Status Messages

IN THIS SECTION

Problem

Description

Review and analyze VPN status messages related to issues caused by an inactive IKE Phase 2.

Symptoms

- IKE Phase 2 is not active.
- The **show security ipsec security-associations** command output does not list the remote address of the VPN.

Solution

The best way to troubleshoot the IKE Phase 2 issues is by reviewing the VPN status messages of the responder firewall.

The responder firewall is the *receiver* side of the VPN that receives the tunnel setup requests. The initiator firewall is the *initiator* side of the VPN that sends the initial tunnel setup requests.

1. Using the CLI, configure a syslog file, **kmd-logs**, for VPN status logs on the responder firewall.

   See KB10097-How to configure syslog to display VPN status messages. As you bring up the VPN tunnel, the messages are captured in **ldm-logs**.

2. Using the CLI, check for Phase 2 error messages: **show log kmd-logs**

   Sample output messages:

-
  ```
  Message:
  Jul 10 16:14:30 210-2 kmd[52472]: IKE Phase-2: Failed to match the peer proxy IDs
  [p2_remote_proxy_id=ipv4_subnet(any:0,[0..7]=192.168.10.0/24),
  p2_local_proxy_id=ipv4_subnet(any:0,[0..7]=10.10.10.0/24)] for local ip: 2.2.2.1, remote
  peer ip:2.2.2.2
  ```

  - Meaning—The proxy identity of the peer device does not match the local proxy identity.

  - Action—The proxy ID must be an exact reverse of the peer's configured proxy ID. See KB10124 - How to fix the Phase 2 error: Failed to match the peer proxy IDs.

-
  ```
  Message:
  Jul 16 21:14:20 kmd[1456]: IKE Phase-2 Failure: Quick mode - no proposal chosen
  [spi=cf0f6152, src_ip=4.4.4.4, dst_ip=3.3.3.2]
  Jul 16 21:14:20 kmd[1456]: KMD_VPN_PV_PHASE2: IKE Phase-2 Failure: Quick mode - no
  proposal chosen [spi=cf0f6152, src_ip=4.4.4.4, dst_ip=3.3.3.2]
  Jul 16 21:14:20 kmd[1456]: IKE Phase-2: Negotiations failed. Local gateway: 4.4.4.4,
  Remote gateway: 3.3.3.2
  ```

  - Meaning—The device running Junos OS did not accept any of the IKE Phase 2 proposals that the specified IKE peer sent.

  - Action—Verify the local Phase 2 VPN configuration elements. The Phase 2 proposal elements include the following:

    - Authentication algorithm

    - Encryption algorithm

    - Lifetime kilobytes

    - Lifetime seconds

    - Protocol

    - Perfect forward secrecy

  You can change the local configuration to accept at least one of the remote peer’s Phase 2 proposals, or contact the remote peer’s administrator and arrange for the IKE configurations at both ends of the tunnel to use at least one mutually acceptable Phase 2 proposal.

Sample output messages:

- **IPsec proposal mismatch**

  -
    ```
    Message:
    Sep 7 09:26:57 kmd[1393]: IKE negotiation failed with error: No proposal chosen. IKE
    Version: 1, VPN: vpn1 Gateway: ike-gw, Local: 10.10.10.1/500, Remote: 10.10.10.2/500,
    Local IKE-ID: 10.10.10.1,
    Remote IKE-ID: 10.10.10.2, VR-ID: 0
    ```

  NOTE: If `Local IKE-ID` and `Remote IKE-ID` are displayed as `Not-Available`, then it is a Phase 1 failure message. See KB30548 - IKE Phase 1 VPN status messages in 12.1X44 and later releases.

  Action—Verify the local Phase 2 VPN configuration elements. The Phase 2 proposal elements include the following:

  - Authentication algorithm
  - Encryption algorithm
  - Lifetime kilobytes
  - Lifetime seconds
  - Protocol
  - Perfect forward secrecy

- **Proxy-ID mismatch**

  Sample output messages:

  -
    ```
    Sep 7 09:23:05 kmd[1334]: IKE Phase-2: Failed to match the peer proxy IDs
    [p2_remote_proxy_id=ipv4_subnet(any:0,[0..7]=192.168.1.0/24),
    p2_local_proxy_id=ipv4_subnet(any:0,[0..7]=192.168.3.0/24)] for local ip: 10.10.10.2,
    remote peer ip:10.10.10.1
    ```

  -
    ```
    Sep 7 09:23:05 kmd[1334]: IKE Phase-2: Failed to match the peer proxy IDs
    [p2_remote_proxy_id=ipv4_subnet(any:0,[0..7]=192.168.1.0/24),
    p2_local_proxy_id=ipv4_subnet(any:0,[0..7]=192.168.3.0/24)] for local ip: 10.10.10.2,
    remote peer ip:10.10.10.1
    ```

  Action—The proxy ID must be an exact reverse match of the peer's configured proxy ID. See KB10124 - How to fix the Phase 2 error: Failed to match the peer proxy IDs.

If the VPN connection is established successfully, you can see the following messages in the syslog:

-
  ```
  Sep 10 08:35:03 kmd[1334]: KMD_PM_SA_ESTABLISHED: Local gateway: 10.10.10.2, Remote
  gateway: 10.10.10.1, Local ID: ipv4_subnet(any:0,[0..7]=192.168.3.0/24), Remote ID:
  ipv4_subnet(any:0,[0..7]=192.168.1.0/24), Direction: inbound, SPI: 0x4b23e914, AUX-SPI: 0,
  Mode: Tunnel, Type: dynamic
  Sep 10 08:35:03 kmd[1334]: KMD_PM_SA_ESTABLISHED: Local gateway: 10.10.10.2, Remote
  gateway: 10.10.10.1, Local ID: ipv4_subnet(any:0,[0..7]=192.168.3.0/24), Remote ID:
  ipv4_subnet(any:0,[0..7]=192.168.1.0/24), Direction: outbound, SPI: 0xa90982b3, AUX-SPI:
  0, Mode: Tunnel, Type: dynamic
  Sep 10 08:35:03 kmd[1334]: KMD_VPN_UP_ALARM_USER: VPN test_vpn from 10.10.10.1 is up.
  Local-ip: 10.10.10.2, gateway name: ike-gw, vpn name: vpn1, tunnel-id: 131073, local
  tunnel-if: st0.0, remote tunnel-ip: Not-Available, Local IKE-ID: 10.10.10.2, Remote IKE-
  ID: 10.10.10.1, XAUTH username: Not-Applicable, VR id: 0
  ```

-
  ```
  Sep 9 06:57:34 kmd[1393]: KMD_PM_SA_ESTABLISHED: Local gateway: 10.10.10.1, Remote
  gateway: 10.10.10.2, Local ID: ipv4_subnet(any:0,[0..7]=192.168.1.0/24), Remote ID:
  ipv4_subnet(any:0,[0..7]=192.168.3.0/24), Direction: inbound, SPI: 0xa90982b3, AUX-SPI: 0,
  Mode: Tunnel, Type: dynamic, Traffic-selector:
  Sep 9 06:57:34 kmd[1393]: KMD_PM_SA_ESTABLISHED: Local gateway: 10.10.10.1, Remote
  gateway: 10.10.10.2, Local ID: ipv4_subnet(any:0,[0..7]=192.168.1.0/24), Remote ID:
  ipv4_subnet(any:0,[0..7]=192.168.3.0/24), Direction: outbound, SPI: 0x4b23e914, AUX-SPI:
  0, Mode: Tunnel, Type: dynamic, Traffic-selector:
  Sep 9 06:57:34 kmd[1393]: KMD_VPN_UP_ALARM_USER: VPN test_vpn from 10.10.10.2 is up. Local-
  ip: 10.10.10.1, gateway name: ike-gw, vpn name: vpn1, tunnel-id: 131073, local tunnel-if:
  st0.0, remote tunnel-ip: Not-Available, Local IKE-ID: 10.10.10.1, Remote IKE-ID:
  10.10.10.2, XAUTH username: Not-Applicable, VR id: 0, Traffic-selector: , Traffic-selector
  local ID: ipv4_subnet(any:0,[0..7]=192.168.1.0/24), Traffic-selector remote ID:
  ipv4_subnet(any:0,[0..7]=192.168.3.0/24)ze: 12px;">IPsec Proposal mismatch
  ```

3. If you could not locate any Phase 2 messages, proceed to Step "4" on page 1431.

4. Using the CLI, review the Phase 2 proposals and confirm that the configuration matches the Phase 2 proposals configured by the peer: **show security ipsec**

```
show security ipsec
proposal ipsec-phase2-proposal {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
}
```

```
policy ipsec-phase2-policy {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals ipsec-phase2-proposal;
}
vpn ike-vpn-srx1 {
    vpn-monitor;
    ike {
        gateway gw-srx1;
        ipsec-policy ipsec-phase2-policy;
    }
}
```

5. If the issue persists, to open a JTAC case with the Juniper Networks support team, see Data Collection for Customer Support for the data you should collect to assist in troubleshooting before opening a JTAC case.

18

CHAPTER

Configuration Statements

aaa

IN THIS SECTION

Syntax

```
aaa {
    access-profile access-profile {
        config-payload-password config-payload-password;
    }
    client {
        password;
        username;
    }
}
```

Hierarchy Level

```
[edit security ike gateway gateway-name]
```

Description

Specify that extended authentication is performed in addition to IKE Phase 1 authentication for remote users trying to access a VPN tunnel. This authentication can be through Extended Authentication (XAuth) or Extensible Authentication Protocol (EAP). Include a previously created access profile, configured with the `edit access profile` statement, to specify the access profile to be used for authentication information.

Options

`access-profile` ***`profile-name`***

Name of the previously created access profile to use for extended authentication for remote users trying to access a VPN.

`config-payload-password`

Specify common client password for IKEv2 configuration payload with 1 to 128 characters.

`client`

Specify an AAA client uername and password for each configured authenticator that is allowed to request authentications for supplicants.

- `password`—AAA client password with 1 to 128 characters.
- `username`—AAA client username with 1 to 128 characters.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 15.1X49-D80.

`config-payload-password` option introduced in Junos OS Release 20.1R1.

RELATED DOCUMENTATION

IPsec Overview | **20**

address-assignment (Access)

IN THIS SECTION

Syntax

```
address-assignment {
    abated-utilization percentage;
    abated-utilization-v6 percentage;
    high-utilization percentage;
    high-utilization-v6 percentage;
    neighbor-discovery-router-advertisement ndra-name;
    pool pool-name {
        family {
            inet {
                dhcp-attributes {
                    boot-file boot-file-name;
                    boot-server boot-server-name;
                    domain-name domain-name;
                    grace-period seconds;
                    maximum-lease-time (seconds | infinite);
                    name-server ipv4-address;
                    netbios-node-type (b-node | h-node | m-node | p-node);
                    next-server next-server-name;
                    option dhcp-option-identifier-code {
                        array {
                            byte [8-bit-value];
                            flag [ false| off |on |true];
                            integer [32-bit-numeric-values];
                            ip-address [ip-address];
                            short [signed-16-bit-numeric-value];
                            string [character string value];
                            unsigned-integer [unsigned-32-bit-numeric-value];
                            unsigned-short [16-bit-numeric-value];
                        }
                        byte 8-bit-value;
                        flag (false | off | on | true);
                        integer 32-bit-numeric-values;
                        ip-address ip-address;
                        short signed-16-bit-numeric-value;
                        string character string value;
                        unsigned-integer unsigned-32-bit-numeric-value;
                        unsigned-short 16-bit-numeric-value;
                    }
                    option-match {
                        option-82 {
                            circuit-id match-value {
                                range range-name;
                            }
                            remote-id match-value;
                                range range-name;
                            }
                        }
                    }
                    propagate-ppp-settings [interface-name];
                    propagate-settings interface-name;
                    router ipv4-address;
                    server-identifier ip-address;
                    sip-server {
                        ip-address ipv4-address;
                        name sip-server-name;
                    }
                    tftp-server server-name;
                    wins-server ipv4-address;
                }
                excluded-address;
                excluded-range range-name
                    high upper-limit;
                    low lower-limit;
                }
                range range-name {
                    high upper-limit;
                    low lower-limit;
                }
                host hostname {
                    hardware-address mac-address;
                    ip-address reserved-address;
                    user-name;
                }
                network network address;
                xauth-attributes {
                    primary-dns ip-address;
                    primary-wins ip-address;
                    secondary-dns ip-address;
                    secondary-wins ip-address;
                }
            }
            inet6 {
                dhcp-attributes {
                    dns-server ipv6-address;
                    grace-period seconds;
                    maximum-lease-time (seconds | infinite);
                    option dhcp-option-identifier-code {
                        array {
                            byte [8-bit-value];
                            flag [ false| off |on |true];
                            integer [32-bit-numeric-values];
                            ip-address [ip-address];
                            short [signed-16-bit-numeric-value];
                            string [character string value];
                            unsigned-integer [unsigned-32-bit-numeric-value];
                            unsigned-short [16-bit-numeric-value];
                        }
                        byte 8-bit-value;
                        flag (false | off | on | true);
                        integer 32-bit-numeric-values;
                        ip-address ip-address;
                        short signed-16-bit-numeric-value;
                        string character string value;
                        unsigned-integer unsigned-32-bit-numeric-value;
                        unsigned-short 16-bit-numeric-value;
                    }
                    propagate-ppp-settings [interface-name];
                    sip-server-address ipv6-address;
                    sip-server-domain-name domain-name;
                }
                excluded-address;
                excluded-range range-name
                    high upper-limit;
                    low lower-limit;
                }
                host hostname {
                    hardware-address mac-address;
                    ip-address reserved-address;
                    user-name;
                }
                prefix ipv6-network-prefix;
                range range-name {
                    high upper-limit;
                    low lower-limit;
                    prefix-length delegated-prefix-length;
                }
                xauth-attributes {
                    primary-dns-ipv6;
                    secondary-dns-ipv6;
                }
            }
        link pool-name;
    }
}
```

Hierarchy Level

```
[edit access]
```

Description

The address-assignment pool feature enables you to create different pools with different attributes. For example, multiple client applications, such as DHCPv4 or DHCPv6, can use an address-assignment pool to provide addresses for their particular clients.

Options

- `host` *`hostname`*—Name by which a network-attached device is known on a network.
- `hardware-address` *`mac-address`*—Specify the MAC address of the client. This is the hardware address that identifies the client on the network.
- `ip-address` *`reserved-address`*—Specify the reserved IP address.
- `user-name`—Specify username or IKE ID.
- `xauth-attributes`—Specify XAuth attributes to use in XAuth authentication.
- `primary-dns-ipv6`—Specify the primary-dns IPv6 address.
- `secondary-dns-ipv6`—Specify the secondary-dns IPv6 address.

Required Privilege Level

access—To view this statement in the configuration.

access-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.4.

`xauth-attributes` option under `inet6` is introduced in Junos OS Release 20.3R1.

`user-name` option under `inet host` *`host-name`* is introduced in Junos OS Release 20.3R1.

`host` option under `inet6` is introduced in Junos OS Release 20.3R1.

advpn

IN THIS SECTION

Syntax

```
advpn {
    suggester {
        disable;
    }
    partner {
        connection-limit number;
        idle-threshold   packets/sec;
        idle-time        seconds;
        disable;
    }
}
```

Hierarchy Level

```
[edit security ike gateway gateway-name]
```

Description

Enable Auto Discovery VPN (ADVPN) protocol on the specified gateway. ADVPN dynamically establishes VPN tunnels between spokes to avoid routing traffic through the Hub.

Options

suggester

VPN peer that can initiate a shortcut exchange to allow shortcut partners to establish dynamic security associations (SAs) with each other. Specify `disable` to disable this role on the gateway.

Both suggester and partner roles are enabled if `advpn` is configured without explicitly configuring `suggester` or `partner` keywords. We do not support suggester and partner roles on the same gateway. You must explicitly configure `disable` with the `suggester` or `partner` keyword to disable that particular role. You cannot disable both suggester and partner roles on the same gateway.

partner

VPN peer that can receive a shortcut exchange suggesting that it should establish dynamic SAs with another peer. Specify `disable` to disable this role on the gateway.

The following options can be configured for the partner role:

connection-limit

Maximum number of shortcut tunnels that can be created with different shortcut partners using a particular gateway. The maximum number, which is also the default, is platform-dependent.

Reducing the configured `connection-limit` value causes all active shortcut tunnels to be brought down. For example, if `connection-limit` is configured as 100 and you later reconfigure the number to 80, all active shortcut tunnels are brought down. Increasing the configured `connection-limit` value does not cause shortcut tunnels to go down.

idle-threshold

Rate, in packets per second, below which the shortcut is brought down.

- **Range:** 3 through 5,000 packets per second.
- **Default:** 5 packets per second.

idle-time

Duration, in seconds, after which the shortcut is deleted if the traffic remains below the `idle-threshold` value.

- **Range:** 60 seconds through 86,400 seconds.
- **Default:** 300 seconds.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.3X48-D10. The range for the `idle-threshold` option and the range and default value for the `idle-time` option revised in Junos OS Release 12.3X48-D20.

RELATED DOCUMENTATION

Understanding Auto Discovery VPN | 919

anti-replay-window-size

IN THIS SECTION

Syntax

```
anti-replay-window-size anti-replay-window-size;
```

Hierarchy Level

```
[edit security ipsec],
[edit tenants name security ipsec]
```

Description

To enable the `anti-replay-window-size` option, you first need to configure the option for each VPN object or at the global level. You can configure the anti-replay window size in the range of 64 to 8192 (power of 2). If the anti-replay window size is not configured, the window size is 64 by default. If `anti-replay-window-size` command is configured at both the global and VPN object levels, the configuration on VPN object takes precedence over global configuration.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 19.2R1.

RELATED DOCUMENTATION

Anti-Replay Window | **186**

application-bypass (Juniper Secure Connect)

IN THIS SECTION

Syntax

```
application-bypass {
    term name {
        description description;
        protocol protocol;
        domain-name domain-name
     }
}
```

Hierarchy Level

```
[edit security remote-access client-config name]
```

Description

Define Juniper Secure Connect remote client configuration parameters for bypassing certain applications based on domain names and protocols without passing through the remote access VPN tunnel. Administrator configures these parameters on the SRX Series Firewall which are pushed to client application after its successful authentication.

Options

application-bypass Define application-bypass configuration.

term Define an application-bypass entry; specify a term *name*.

- Values:
  - Format: Must be a string beginning with a number or letter and consisting of letters, numbers, dashes and underscores. Supports upto 50 term entries.

description Text description of remote-access application-bypass profile

domain-name Specify domain-name to bypass VPN tunnel.

- Values:
  - `contains`—Match any domain name containing the suffix value. Example: *example.com*.
  - `fqdn`—Match fully qualified domain name. Example: *hr.example.com*.
  - `wildcard`—Match any subdomain. Example: *.example.com*.

protocol Specify protocol to bypass VPN tunnel

- Values:
  - `all`—Bypass both TCP and UDP traffic.
  - `tcp`—Bypass only TCP traffic.
  - `udp`—Bypass only UDP traffic.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 23.1R1.

RELATED DOCUMENTATION

Juniper Secure Connect Administration Guide

authentication-order (Access Profile)

IN THIS SECTION

Syntax

```
authentication-order [ldap | password | radius | securid];
```

Hierarchy Level

```
[edit access profile profile-name]
```

Description

Set the order in which the Junos OS tries different authentication methods when verifying that a client can access the devices. For each login attempt, the software tries the authentication methods in order, from first to last.

Options

- `ldap`—Verify the client using LDAP.
- `password`—Verify the client using the information configured at the `[edit access profile` *`profile-name`* `client` *`client-name`*`]` hierarchy level.
- `radius`—Verify the client using RADIUS authentication services.
- `securid`—Verify the client using SecurID authentication services.

Required Privilege Level

access—To view this statement in the configuration.

access-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 9.1.

RELATED DOCUMENTATION

Example: Specifying RADIUS Server Connections on a Security Device

Ethernet Switching User Guide

auto-re-enrollment (Security)

IN THIS SECTION

Syntax

```
auto-re-enrollment {
    cmpv2 {
        certificate-id certificate-id-name {
            ca-profile-name ca-profile-name;
            challenge-password password;
            re-enroll-trigger-time-percentage percentage;
            re-enroll-time (days value| hours value| percentage value);
            re-generate-keypair;
        }
    }
    scep {
        certificate-id certificate-id-name {
            ca-profile-name ca-profile-name;
            challenge-password password;
            re-enroll-trigger-time-percentage percentage;
            re-enroll-time (days value| hours value| percentage value);
            re-generate-keypair;
            scep-digest-algorithm {
            (md5 | sha1);
             }
            scep-encryption-algorithm {
            (des | des3);
             }
        }
    }
}
```

Hierarchy Level

```
[edit security pki]
```

Description

Configure the automatic reenrollment of a local end-entity (EE) certificate. Auto-reenrollment requests that the issuing CA replace a device certificate before its specified expiration date.

Options

certificate-id Auto reenrollment configuration for certificate ID.

ca-profile-name Specify the name of the certificate authority (CA) profile to be used for automatic reenrollment. The CA certificate must be present to initiate reenrollment.

challenge-password Specify the password used by the certificate authority (CA) for enrollment and revocation. If the CA does not provide the challenge password, choose your own password.

re-enroll-trigger-time-percentage Specify the certificate reenrollment trigger as a percentage of the end-entity (EE) certificate’s lifetime that remains before certificate reenrollment is initiated. For example, if the renewal request is to be sent when the certificate's remaining lifetime is 10 percent, then configure 10 for `re-enroll-trigger-time-percentage` value. The time at which the certificate reenrollment is initiated is based on the certificate expiry date.

- **Range:** 1 through 99

re-enroll-time

This option allows you to trigger auto-re-enrollment ahead of the certificate expiration. You can configure the re-enrollment trigger time in days, or hours, or percentage.

- days *value*—Specify when to trigger re-enrollment in days.
- hours *value*—Specify when to trigger re-enrollment in hours.
- percentage *value*—Specify when to trigger re-enrollment in percentage. **Range**: 1 to 99.

If you configure both `re-enroll-trigger-time-percentage` and `re-enroll-time` options, then `re-enroll-time` configuration take precedence.

Starting Junos OS Release 23.1R1, you must configure either `re-enroll-trigger-time-percentage` or `re-enroll-time` for the `commit check` to be successful.

re-generate-keypair

Specify new key pair generation for automatic certificate reenrollment. If this statement is not configured, the current key pair is used. If the key pair does not change, the CA does not issue new certificates. We recommend that a new key pair be generated during reenrollment as it provides better security.

scep-digest-algorithm

SCEP digest algorithm.

- Values:
  - md5—Use MD5 as SCEP digest algorithm
  - sha1—Use SHA1 as SCEP digest algorithm

scep-encryption-algorithm

SCEP encryption algorithm.

- Values:
  - des—Use DES as SCEP encryption algorithm
  - des3—Use DES3 as SCEP encryption algorithm

cmpv2

Configure automatic reenrollment of a local certificate using CMPv2.

scep

Configure automatic reenrollment of a local certificate using Simple Certificate Enrollment Protocol (SCEP).

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 9.0. `cmpv2` and `scep` options added in Junos OS Release 15.1X49-D40.

Support for `re-enroll-time (days` *`value`*`| hours` *`value`*`| percentage` *`value`*`)` option added in Junos OS Release 21.4R1.

RELATED DOCUMENTATION

PKI Components In Junos OS | 33

show security ipsec statistics (MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall) | 1997

ca-profile (Security PKI)

IN THIS SECTION

Syntax

```
ca-profile ca-profile-name {
    administrator {
        e-mail-address e-mail-address;
    }
    ca-identity ca-identity ;
    enrollment {
            retry number;
            retry-interval seconds;
            url url-name;
    }
    proxy-profile;
    revocation-check;
    routing-instance routing-instance-name ;
    source-address ip-address;
}
```

Hierarchy Level

```
[edit security pki]
```

Description

Configure certificate authority (CA) profile. The CA profile contains the name and URL of the CA or RA, as well as retry-timer settings.

Options

ca-profile-name Name of a trusted CA.

administrator *email-address*

Specify an administrator e-mail address to which the certificate request is sent. By default, there is no preset e-mail address.

ca-identity

Specify the certificate authority (CA) identity to use in requesting digital certificates. This name is typically the domain name of the CA.

enrollment

Specify the enrollment parameters for a certificate authority (CA).

retry *number*

Number of automated attempts for online enrollment to be retried in case enrollment response is pending.

- **Range:** 0 through 1080
- **Default:** 10

retry-interval *seconds*

Time interval between the enrollment retries.

- **Range:** 0 through 3600
- **Default:** 900 seconds

url *url-name*

Enrollment URL where the Simple Certificate Enrollment Protocol (SCEP) or CMPv2 request is sent to the certification authority (CA) as configured in this profile. With SCEP, you enroll CA certificates with the `request security pki ca-certificate enroll` command and specify the CA profile. There is no separate command to enroll CA certificates with CMPv2. The IP address in the enrollment URL can be an IPv4 or an IPv6 address.

proxy-profile

Use specified proxy server. If proxy profile is configured in CA profile, the device connects to the proxy host instead of the CA server while certificate enrollment, verification or revocation. The proxy host communicates with the CA server with the requests from the device, and then relay the response to the device.

Public key infrastructure (PKI) uses proxy profile configured at the system-level. The proxy profile being used in the CA profile must be configured at the `[edit services proxy]` hierarchy. There can be more than one proxy profile configured under `[edit services proxy]` hierarchy. Each CA profile is referred to the most one such proxy profile. You can configure host and port of the proxy profile at the `[edit system services proxy]` hierarchy.

revocation-check

Specify the method the device uses to verify the revocation status of digital certificates.

routing-instance

Specify the routing-instance to be used.

source-address Specifies a source IPv4 or IPv6 address to be used instead of the IP address of the egress interface for communications with external servers. External servers are used for certificate enrollment and reenrollment using Simple Certificate Enrollment Protocol (SCEP) or Certificate Management Protocol version 2 (CMPv2), downloading certificate revocation lists (CRLs) using HTTP or LDAP, or checking certificate revocation status with Online Certificate Status Protocol (OCSP). If this option is not specified then the IP address of the egress interface is used as the source address.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5. Support for `ca-identity` option is added in Junos OS Release 11.1. Support for `ocsp` and `use-ocsp` options added in Junos OS Release 12.1X46-D20.

Support for `proxy-profile` option is added in Junos OS Release 18.2R1.

Support for `source-address` is introduced in Junos OS Release 15.1X49-D60.

RELATED DOCUMENTATION

Basic Elements of PKI in Junos OS

certificate (Juniper Secure Connect)

IN THIS SECTION

Syntax

```
certificate {
    no-expiry-warning;
    no-pin-request-per-connection;
    warn-before-expiry days;
}
```

Hierarchy Level

```
[edit security remote-access client-config]
```

Description

Define certificate identifier parameters for Juniper Secure Connect.

Options

no-expiry-warning	Disable certificate expiry warning.
no-pin-request-per-connection	Disable certificate pin request per connection.

warn-before-expiry	Enable certificate expiration warning in days before the expiry date. • **Default:** 60 days • **Range:** 1 through 90

Required Privilege Level

security

Release Information

Statement introduced in Junos OS Release 20.3R1.

RELATED DOCUMENTATION

Juniper Secure Connect Administrator Guide

certificate

IN THIS SECTION

Syntax

```
certificate {
    local-certificate certificate-id;
    peer-certificate-type (pkcs7 | x509-signature);
    policy-oids oid;
    trusted-ca {
        ca-profile ca-profile-name;
        trusted-ca-group trusted-ca-group-name;
    }
}
```

Hierarchy Level

```
[edit security ike policy policy-name]
```

Description

Specify usage of a digital certificate to authenticate the virtual private network (VPN) initiator and recipient.

Options

`local-certificate` *`certificate-id`* —Specify a particular certificate when the local device has multiple loaded certificates. The device deletes existing IKE and IPsec SAs when you update the `local-certificate` configuration in the IKE policy. Starting in Junos OS Release 19.1R1, a commit check is added to prevent user from adding `.`, `/`, `%`, and space in a certificate identifier while generating a local or remote certificates or a key pair.

`peer-certificate-type`—Specify a preferred type of certificate (PKCS7 or X509).

- `pkcs7`—Public-Key Cryptography Standard #7.
- `x509-signature`—X509 is an ITU-T standard for public key infrastructure. This is the default value.

`policy-oids` *`oid`*—Configure policy object identifiers (OIDs). This configuration is optional. Policy OID contained in a peer's certificate or certificate chain. Up to five policy OIDs can be configured. Each OID can be up to 63 bytes long. You must ensure that at least one of the configured policy OIDs is included in a peer's certificate or certificate chain. Note that the **policy-oids** field in a peer's certificate is optional. If you configure policy OIDs in an IKE policy and the peer's certificate chain does not contain any policy OIDs, certificate validation for the peer fails.

`trusted-ca`—Specify a name for the trusted CA group. A minimum of one CA profile is mandatory to create a trusted CA group and a maximum of 20 CAs are allowed in one trusted CA group. Any CA from a particular group can validate the certificate for that particular entity. Specify the preferred certificate authority (CA) to use when requesting a certificate from the peer. You can associate an IKE policy to a single trusted CA profile or a trusted CA group. During certificate validation the IKE policy will limit itself to the configured group of CAs while establishing a secure connection. Any certificate issued other than the single trusted CA or the trusted CA group are not validated.

- `ca-profile` *`ca-profile-name`*—Specify a name for the CA profiles. A Certificate Authority (CA) is an entity that issues digital certificates which helps to establish secure connection between peers through certificate validation.

- `trusted-ca-group` *`trusted-ca-group-name`*—Specify a name for the trusted CA group. A minimum of one CA profile is mandatory to create a trusted CA group and a maximum of 20 CAs are allowed in one trusted CA group. Any CA from a particular group can validate the certificate for that particular topology.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5. `policy-oids` option added in Junos OS Release 12.3X48-D10. Support for `trusted-ca` option added in Junos OS Release 18.1R1.

RELATED DOCUMENTATION

IPsec Overview | 20

Understanding Certificate Authority Profiles | 46

client-config (Juniper Secure Connect)

IN THIS SECTION

Syntax

```
client-config name {
    application-bypass {
        term name {
            description description;
            protocol protocol;
            domain-name domain-name;
     }
}
biometric-authentication;
    domain-name domain-name;
    certificate {        no-expiry-warning;
        no-pin-request-per-connection;
        warn-before-expiry days;
    }
    connection-mode (always | manual);
    credentials (username | password);
    dead-peer-detection {
        interval seconds;
        threshold threshold;
    }
    no-dead-peer-detection;
    no-eap-tls;
    no-tcp-encap;
    windows-logon {
        auto-dialog-open;
        disconnect-at-logoff;
        domain domain;
        eap-auth;
        flush-credential-at-logoff;
        lead-time-duration seconds;
        mode (automatic | manual);
    }
}
```

Hierarchy Level

```
[edit security remote-access]
```

Description

Define Juniper Secure Connect remote client configuration parameters. The parameters define how Juniper Secure Connect client establishes VPN tunnel with your security device.

Options

name Name of configuration object name.

application-bypass Define application-bypass configuration.

biometric-authentication Enable biometric authentication.

domain-name Define the set of search domain name. As a system administrator, you can configure the set of search domain name that the Juniper Secure Connect application will use to handle DNS lookups. This is applicable to both full tunnels and split tunnel configurations.

You can provide more than one search domain names by executing the `set security remote-access client-config name domain-name domain-name` multiple times. When you enter more than one domain name, it automatically adds a separator (comma) to that value. The number of domain names are limited to the total number of characters and must not exceed 1023 characters. For example, the two domain names `juniper.net,lab.juniper.net` consumes 27 characters while `juniper.net` consumes 11 characters.

Range: 0-1023 characters including comma.

connection-mode

Set one of the following connection mode for clients:

- Values:
  - always—Connect to the VPN automatically when user logs in to remote client device. In always mode, the first VPN connection established when the user clicks the "Connect" button. After that, whenever VPN connection gets disconnected without manual intervention, the client device always attempts to re-establish the connection automatically.
  - manual—Connect to the VPN manually.
- **Default:** manual

credentials

Set one of the following to save the user credentials in Juniper Secure Connect application:

Values:

- username—To save the username in Juniper Secure Connect application. When you enable this option, user will not be required to provide username every time they connect to Juniper Secure Connect application.
- password—To save both the username and password in Juniper Secure Connect application. When you enable this option, user will not be required to provide username and password every time they connect to Juniper Secure Connect application.

Note that you cannot configure both the options at the same time. If you have not configured the `credentials` configuration options, then the Juniper Secure Connect application does not remember any user credentials.

dead-peer-detection—

Enable dead-peer-detection on the client.

Interval The time between DPD probe messages in seconds.

- **Default:** 60 seconds

threshold — Maximum number of DPD retransmissions.

- **Default:** 5

no-dead-peer-detection — Disable dead-peer-detection on client

no-eap-tls — Disable EAP-TLS IKEV2 method.

no-tcp-encap — Disable tcp encapsulation.

windows-logon — Specify windows logon options.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security

Release Information

Statement introduced in Junos OS Release 20.3R1.

Support for `domain-name` option at the `[edit security remote-access client-config name]` hierarchy level added in Junos OS Release 22.1R1.

Support for `application-bypass` option added in Junos OS Release 23.1R1.

RELATED DOCUMENTATION

Juniper Secure Connect Administrator Guide

compliance (Juniper Secure Connect)

IN THIS SECTION

Syntax

```
compliance
    pre-logon name {
        term term-name {
             match {
                 platform {
                    (android | ios | macos | windows) {
                        (app-version | os-version) {
                            (equal | greater-than | greater-than-or-equal | less-than | less-
than-or-equal) version;
                    }
                }
            }
                hostname value;
                ms-domain value;
                ms-workgroup value;
                deviceid value;
            }
             action (accept | reject);
    }
}
```

Hierarchy Level

```
[edit security remote-access]
```

Description

The statement defines Juniper Secure Connect remote-access prelogon compliance policies. You associate a single compliance rule object per remote-access connection profile. This means, a remote-access connection profile can have one associated compliance policy. The Juniper Secure Connect application sends details to the SRX Series Firewall. The device performs prelogon compliance checks. Based on the prelogon compliance rule match, action is taken to accept or reject a connection.

You can create multiple prelogon compliance policies and each policy can contain multiple term rules. The term rules are a set of individual rules containing match conditions and their actions based on the compliance parameters listed in the options below. You can associate a single compliance rule object per remote-access connection profile.

Evaluation Criteria

For every connection request, SRX Series Firewall processes each rule as follows –

1. SRX Series Firewall evaluates the term rules in the order they appear in the configuration.
2. If there is no match in the current term rule, it evaluates the next term rule.
3. Based on the match, it takes an action.
4. When there is no action specified, the default action for a match rule is `reject`.
5. When no further term rule is specified for an unmatched rule, the default action is `reject`
6. When no compliance rule is attached to the profile, the default action is `accept`.

Based on this evaluation criteria, the administrator defines rules.

Options

pre-logon Define pre-login compliance rule; specify the compliance rule *name*.

- Values:
  - Format: String beginning with a number or letter and consisting of letters, numbers, dashes and underscores.
  - Range: Supports upto 255 prelogon compliance rules.

term Define compliance rule term; specify the term rule *name*.

- Values:
  - Format: Must be a string beginning with a number or letter and consisting of letters, numbers, dashes and underscores.
  - Range—Supports upto 10 term rules per compliance rule.

action Specify the action based on the rule match.

- Values:
  - `accept`—To approve the request.
  - `reject`—To reject the request. This is the default action for an unmatched rule when no further term rule is specified.
  - Default: When no action is specified, the default action is `reject`.

match Specify rules to match.

platform Specify rule to match OS and Client information for the specified OS.

- Values:
  - `android`—Specify android version and app-version.
  - `ios`—Specify iOS version and app-version.
  - `macos`—Specify macOS version and app-version.
  - `windows`—Specify Windows OS version and app-version.

app-version Match remote access client version with the specified operational values.

- Values:

- `equal`—Perform operation 'equal'.
- `greater-than`—Perform operation 'greater-than'.
- `greater-than-or-equal`—Perform operation 'greater-than-or-equal'.
- `less-than`—Perform operation 'less-than'.
- `less-than-or-equal`—Perform operation 'less-than-or-equal.

os-version

Match operating system version with the specified operational values.

- Values:
  - `equal`—Perform operation 'equal'.
  - `greater-than`—Perform operation 'greater-than'.
  - `greater-than-or-equal`—Perform operation 'greater-than-or-equal'.
  - `less-than`—Perform operation 'less-than'.
  - `less-than-or-equal`—Perform operation 'less-than-or-equal.

version

Specify version.

- Values:
  - `app-version` supports numeric and . (period) characters.
  - `os-version` supports alphanumeric and . (period) characters.
- Range:
  - `app-version` supports upto 16 entries per term rule.
  - `os-version` supports upto 16 entries per term rule.

deviceid

Specify set of device IDs.

- Values:
  - Supports alphanumeric, +, /, and = characters.

    This is a list of values.
- Range:

- Supports upto 1024 entries per term.

hostnames

Specify set of host names.

- Values:
  - Supports alphanumeric, - and _ characters.

    This is a list of values.
- Range:
  - Supports upto 1024 entries per term rule.

ms-domain

Specify set of domain names.

- Values:
  - Supports alphanumeric, - and _ characters.

    This is a list of values.
- Range:
  - Supports upto 16 entries per term rule.

ms-workgroup

Specify set of work groups.

- Values:
  - Supports alphanumeric, - and _ characters.

    This is a list of values.
- Range:
  - Supports upto 16 entries per term rule.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 23.1R1.

RELATED DOCUMENTATION

Juniper Secure Connect Administration Guide

crl (Security)

IN THIS SECTION

Syntax

```
crl {
    disable {
        on-download-failure;
    }
    refresh-interval hours;
    url url-name;
}
```

Hierarchy Level

```
[edit security pki ca-profile ca-profile-name revocation-check]
```

Description

Configure the certificate revocation list (CRL). A CRL is a time-stamped list identifying revoked certificates, which is signed by a CA and made available to the participating IPsec peers on a regular periodic basis.

Options

disable on-download-failure

(Optional) Override the default behavior and permit certificate verification even if the CRL fails to download.

refresh-interval *hours*

Specify the amount of time interval in hours between certificate revocation list (CRL) updates.

- **Range:** 0 through 8784 hours.

  Configuring `refresh-interval` value as 0 or not configuring `refresh-interval` is considered as same in Junos. In both the cases, CRL is updated based on the value specified for the next-update time in the received CRL.

- **Default:** The CRL is updated based on the value specified for the next-update time in the received CRL. This update occurs in the following cases:

  - if the `refresh-interval` is not configured.

  - if the `refresh-interval` value is configured as 0.

url *url-name*

Name of the location from which to retrieve the CRL through HTTP or Lightweight Directory Access Protocol (LADP). You can specify one URL for each configured CA profile. By default, no location is specified. Use a fully qualified domain name (FQDN) or an IP address and, optionally, a port number. If no port number is specified, port 80 is used for HTTP and port 443 is used for LDAP.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5.

`disable` option is introduced in Junos OS Release 9.0.

RELATED DOCUMENTATION

PKI Components In Junos OS | 33

revocation-check (Security PKI) | 1618

dead-peer-detection

IN THIS SECTION

Syntax

```
dead-peer-detection {
    (always-send | optimized | probe-idle-tunnel);
    interval seconds;
    threshold number;
}
```

Hierarchy Level

```
[edit security ike gateway gateway-name]
```

Description

Enable the device to use dead peer detection (DPD). DPD is a method used by devices to verify the current existence and availability of IPsec peers. A device performs this verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE messages) to a peer and waiting for DPD acknowledgements (R-U-THERE-ACK messages) from the peer.

Options

interval Specify the amount of time that the peer waits for traffic from its destination peer before sending a dead-peer-detection (DPD) request packet.

- **Default:** 10 seconds
- **Range:** 2 through 60 seconds

always-send Instructs the device to send dead peer detection (DPD) requests regardless of whether there is outgoing IPsec traffic to the peer.

optimized
Send dead peer detection (DPD) messages if there is no incoming IKE or IPsec traffic within the configured interval after outgoing packets are sent to the peer. This is the default DPD mode.

probe-idle-tunnel
Send dead peer detection (DPD) messages during idle traffic time between peers.

threshold
Specify the maximum number of unsuccessful dead peer detection (DPD) requests to be sent before the peer is considered unavailable.

- **Default:** 5
- **Range:** 1 through 5

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5. Support for the `optimized` and `probe-idle-tunnel` options added in Junos OS Release 12.1X46-D10.

RELATED DOCUMENTATION

Understanding AutoVPN

IPsec VPN Overview

dead-peer-detection (Security Group VPN Server)

IN THIS SECTION

Syntax

```
dead-peer-detection {
    always-send;
    interval seconds;
    threshold number;
}
```

Hierarchy Level

```
[edit security group-vpn server ike gateway gateway-name]
```

Description

Enable the device to use dead peer detection (DPD). DPD is a method used by devices to verify the current existence and availability of IPsec peers. A device performs this verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE messages) to a peer and waiting for DPD acknowledgements (R-U-THERE-ACK messages) from the peer.

Options

`always-send`—Send probes periodically regardless of incoming and outgoing data traffic.

`interval` *seconds*—Specify the interval time in seconds between DPD probe messages.

- **Range:** 10 through 60 seconds
- **Default:** 10 seconds

`threshold` *number*—Specify the maximum number of DPD retransmissions.

- **Range:** 1 through 5
- **Default:** 5

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Support for the Group VPN server added in Junos OS Release 15.1X49-D30 for vSRX Virtual Firewall.

RELATED DOCUMENTATION

Group VPNv2 Overview | **758**

gateway (Security Group VPN Server IKE) | **1503**

decryption-failures

IN THIS SECTION

Syntax

```
decryption-failures {
    threshold value;
}
```

Hierarchy Level

```
[edit security alarms potential-violation]
```

Description

Raise a security alarm after exceeding a specified number of decryption failures.

Default

Multiple decryption failures do not cause an alarm to be raised.

Options

failures—Number of decryption failures up to which an alarm is not raised. When the configured number is exceeded, an alarm is raised.

- **Range:** 1 through 1,000,000,000.
- **Default:** 1000

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 11.2.

RELATED DOCUMENTATION

IPsec Overview | **20**

Monitoring VPN Traffic | **1351**

default-profile (Juniper Secure Connect)

IN THIS SECTION

Syntax

```
default-profile default-profile;
```

Hierarchy Level

```
[edit security remote-access]
```

Description

Configure default profile. On your security device, you must specify one of the remote-access profiles as the default profile.

NOTE: Starting in Junos OS Release 23.1R1, we've hidden the `default-profile` option at the [`edit security remote-access`] hierarchy level. In releases before Junos OS Release 23.1R1, you use this option to specify one of the remote-access profiles as the default profile in Juniper Secure

Connect. But with changes to the format of remote-access profile names, we no longer require the `default-profile` option.

We've deprecated `default-profile` option—rather than immediately removing it—to provide backward compatibility and a chance to make your existing configuration conform to the changed configuration. You'll receive a warning message if you continue to use the `default-profile` option in your configuration. However existing deployments are not affected if you modify the current configuration. See profile (Juniper Secure Connect).

In existing deployments, to ensure a smooth transition with this change, we recommend that you modify the current configuration with profile *hr* to *ra.example.com/hr* or *192.168.1.10/hr* at the `[edit]` hierarchy using below commands -

-
  ```
  user@host# rename security remote-access profile hr to profile ra.example.net/hr
  ```

-
  ```
  user@host# rename edit security remote-access profile hr to profile 192.168.1.10/hr
  ```

For new configurations, consider the following scenarios to create a new remote-access profile based on how your end users connect using the Juniper Secure Connect application -

- If your end users connect using an *IP address*, specify the *IP address* in the profile name.
- If your end users connect using *FQDN*, specify the *FQDN* in the profile name.
- If you need to separate users with different realm values such as *hr*, append */hr* to the *IP address* or *FQDN* as follows -
  - `[edit security remote-access profile ra.example.net/hr]`
  - `[edit security remote-access profile 192.168.1.10/hr]`

Required Privilege Level

security

Release Information

Statement introduced in Junos OS Release 20.3R1

The *default-profile* option is not available in SRX Series Firewalls starting Junos OS Release 23.1R1.

RELATED DOCUMENTATION

Juniper Secure Connect Administrator Guide

default-trusted-ca-certs (Security)

IN THIS SECTION

Syntax

```
default-trusted-ca-certs {
    automatic-download {
        deactivate;
        interval {
             hours value;
        }
        url value;
        routing-instance value;
     }
  }
```

Hierarchy Level

```
[edit security pki]
```

Description

Dynamic update of trusted CA bundle requires -

- Downloading of trusted CA bundle from Juniper Networks security website, https://signatures.juniper.net/cacert or a custom URL.
- Uploading trusted CA bundle to PKI.
- Periodic polling of trusted CA bundle.

This functionality is configured using the statement `default-trusted-ca-certs`.

Options

automatic-download Sets automatic download of CA certs configuration.

deactivate Disables automatic download of default CA certs.

Use this option when automatic download is configured and you plan to disable it.

interval Specify default trusted CA certs automatic download interval.

- Value:
  - `hours`—Specify a value between 1-336 hours.
- Default:
  - If nothing is specified, it considers 24 hours as the interval.

routing-instance Specify a routing instance for trusted CA cert download. Use this option to configure non-default routing instance.

- Value:
  - *routing-instance-name*. Example: R1

url Specify HTTP URL for OCSP (Online Certificate Status Protocol) access location. This option sets the base URL for downloading trusted CA certs.

- Value:
  - *url*.
- Default:
  - If nothing is specified, default Juniper CDN server URL http://signatures.juniper.net/cacert is considered.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 23.2R1.

RELATED DOCUMENTATION

pki

Dynamic Updated of CA Bundle

dh-group (Security IKE)

IN THIS SECTION

Syntax

```
dh-group (group1 | group2 | group5 | group14 | group15 | group16 | group19 | group20 | group21 |
group24);
```

Hierarchy Level

```
[edit security ike proposal proposal-name]
```

Description

Specify the IKE Diffie-Hellman group. The device does not delete existing IPsec SAs when you update the `dh-group` configuration in the IKE proposal.

Options

dh-group—Diffie-Hellman group for key establishment.

- group1—768-bit Modular Exponential (MODP) algorithm.
- group2—1024-bit MODP algorithm.
- group5—1536-bit MODP algorithm.
- group14—2048-bit MODP group.
- group15—3072-bit MODP algorithm.
- group16—4096-bit MODP algorithm.
- group19—256-bit random Elliptic Curve Groups modulo a Prime (ECP groups) algorithm.
- group20—384-bit random ECP groups algorithm.
- group21—521-bit random ECP groups algorithm.
- group24—2048-bit MODP Group with 256-bit prime order subgroup.

We recommend that you use group14, group15, group16, group19, group20, or group21 instead of group1, group2, or group5.

We support group15, group16, and group21 options only with iked process when junos-ike package is installed.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5.

Support for the group14 option added in Junos OS Release 11.1.

Support for group19, group20, and group24 options added in Junos OS Release 12.1X45-D10.

Support for `group19` and `group20` options added in Junos OS Release 15.1X49-D70 for vSRX Virtual Firewall.

Support for `group15`, `group16`, and `group21` options added in Junos OS Release 19.1R1 on SRX5000 line with `junos-ike` package installed.

Starting in Junos OS Release 20.2R1, we've changed the help text description as `NOT RECOMMENDED` for the CLI options `group1`, `group2`, and `group5` for devices running IKED with `junos-ike` package installed.

Support for `group15`, `group16`, and `group21` options added in Junos OS Release 20.3R1 on vSRX Virtual Firewall instances with `junos-ike` package installed.

Support for `group15`, `group16`, and `group21` options added in Junos OS Release 21.1R1 on vSRX Virtual Firewall 3.0 instances with `junos-ike` package installed.

RELATED DOCUMENTATION

IPsec Overview | 20

proposal (Security IKE) | 1599

Installing Junos IKE package

distinguished-name (Security)

IN THIS SECTION

Syntax

```
distinguished-name <container container-string> <wildcard wildcard-string>
```

Hierarchy Level

```
[edit security ike gateway gateway-name dynamic]
```

Description

Specify a distinguished name as the identifier for the remote gateway with a dynamic IP address.

Options

container-string DN field and value to be matched. For example, `cn`=admin, `ou`=eng, `o`=example, `dc`=net. Specify one or more distinguished name (DN) field and value pairs that must match the DN in the VPN peer's digital certificate. The order of the fields and their values must exactly match the DN in the peer's digital certificate.

Add a space between each field and value pair. For example, `edit security ike gateway jsr_gateway dynamic distinguished-name container o=example, dc=net`.

wildcard-string DN field and value pairs to be matched. For example, `cn`=admin, `ou`=eng, `o`=example, `dc`=net. Specify one or more distinguished name (DN) field and value pairs that must match the DN in the VPN peer's digital certificate. The configured field and value must match the DN in the peer's digital certificate but the order of the fields in the DN does not matter.

Add a space between each field and value pair. For example, `edit security ike gateway jsr_gateway dynamic distinguished-name wildcard o=example, dc=net`.

Starting in Junos OS Release 19.4R1, you can now configure only one dynamic DN attribute among `container-string` and `wildcard-string` at `[edit security ike gateway` *`gateway_name`* `dynamic distinguished-name]` hierarchy. If you try configuring the second attribute after you configure the first attribute, the first attribute is replaced with the second attribute. Before your upgrade your device, you must remove one of the attributes if you have configured both the attributes.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5.

RELATED DOCUMENTATION

IPsec Overview | 20

distribution-profile

IN THIS SECTION

Syntax

```
distribution-profile (fat-core | name) {
    description description;
    fpc fpc {
        pic pic {
            thread-id thread-id;
        }
    }
}
```

Hierarchy Level

```
[edit security]
```

Description

The `distribution-profile` option is introduced to give the administrator an option to define a profile to handle tunnels associated with a certain VPN object. If the default profiles such as `default-spc3-profile` or `default-spc2-profile` are not selected, a new user-defined profile can be created. In a profile, you should mention the Flexible PIC Concentrator (FPC) slot and the PIC slot. When this profile is associated with a VPN object, all matching tunnels will be distributed across these PICs. The `thread-id` is an optional value. If you specify a thread ID, then the tunnel is distributed in the specified thread.

Starting in Junos OS Release 20.2R1, when you add, change, or delete the thread ID from distribution profile, all tunnels part of modified distribution profile anchored on modified SPU member of distribution profile are teared down and re-negotiated. See Table 124 on page 1492 for catastrophic changes when you change the distribution profile configuration.

Table 124: Distribution Profile Catastrophic Change

Distribution Profile Change	Catastrophic Change
Add new distribution profile added to the VPN.	All tunnels part of this new distribution profile are brought down and re-negotiated.
Add new SPU information to a profile already part of the VPN.	No impact on any tunnel.
Delete SPU information from a distribution profile of the VPN.	Only those tunnels part of the distribution profile that is modified and anchored on a deleted SPU are brought down and re-negotiated.
Add first thread ID to an SPU part of the distribution profile.	All tunnels part of this distribution profile are brought down and re-negotiated.
Add next set of thread IDs to SPU part of the distribution profile.	No impact for any tunnel.
Delete a thread ID from an SPU part of the distribution profile.	Only those tunnels part of the modified distribution profile and anchored on the deleted SPU are brought down and re-negotiated.
Delete last thread ID from SPU part of the distribution profile.	No impact on any tunnel.
Delete distribution profile from the VPN	No impact on any tunnel.
Change distribution profile name from profileA to profileB in VPN.	All tunnels part of this profile are brought down and re-negotiated.

Options

description Text description of the distribution profile.

fpc	FPC slot number.
pic	PIC slot number.
thread-id	(Optional) Thread ID number. Only valid for SPC3.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 19.2R1.

dynamic (Security)

IN THIS SECTION

Syntax

```
dynamic {
    connections-limit number;
    distinguished-name {
        container container-string;
        wildcard wildcard-string
    }
    general-ikeid;
    hostname domain-name;
    ike-user-type (group-ike-id | shared-ike-id);
    inet ip-address;
    inet6 ipv6-address;
    reject-duplicate-connection;
    user-at-hostname e-mail-address;
}
```

Hierarchy Level

```
[edit security ike gateway gateway-name]
```

Description

Specify the identifier for the remote gateway with a dynamic IPv4 or IPv6 address. Use this statement to set up a VPN with a gateway that has an unspecified IPv4 or IPv6 address.

Options

connections-limit

Configure the number of concurrent connections that the group profile supports. When the maximum number of connections is reached, no more dynamic virtual private network (VPN) endpoints dialup users attempting to access an IPsec VPN are allowed to begin Internet Key Exchange (IKE) negotiations. This configuration applies to SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX instances, and to SRX5400, SRX5600, and SRX5800 devices configured for AutoVPN.

distinguished-name Specify a distinguished name as the identifier for the remote gateway with a dynamic IP address.

general-ikeid Disables IKE ID validation. If this option is enabled, the new iked process skips the IKE ID validation. After skipping the IKE ID validation, the new iked process still continues the authentication as per the IKE standard. `general-ikeid` is an optional configuration statement.

hostname Name by which a network-attached device is known on a network. A fully qualified domain name (FQDN), or partial FQDN that can be matched to a peer's X.509 PKI certificate. A partial FQDN is matched to the right-most part of the alternate subject field in the peer device's certificate. For example, the partial FQDN example.net can match devices with host1.example.net or host2.example.net in the alternate subject field of their certificates. Note that the partial FQDN example.net does not match host1.example.network.com or host2.net.com because example.net is not the right-most value in the alternate subject field. For AutoVPN, a partial FQDN combined with ike-user-type group-ike-id can be used to identify a specific remote user or peer when there are multiple peers that share a common domain name.

ike-user-type Configure the type of IKE user for a remote access connection.

- Values:
  - group-ike-id—E-mail address or fully qualified domain name (FQDN) shared by a group of remote access users so that each user does not need to configure a separate IKE profile. When group IKE IDs are configured, the IKE ID of each user is a concatenation of a user-specific part and a part that is common to all group IKE ID users. For example, the user Bob might use ”Bob.example.net“ as his full IKE ID, where ”.example.net“ is common to all users. The full IKE ID is used to uniquely identify each user connection. Group IKE IDs require the generation of a unique preshared key based on the username supplied during VPN connection, which can be viewed with the `show security ike pre-shared-key` command.
  - shared-ike-id—E-mail address shared by a large number of remote access users so that each user does not need to configure a separate IKE profile. When a shared IKE ID is configured, all users share a single IKE ID and a single IKE

preshared key. Each user is authenticated through the mandatory XAuth phase, where the credentials of individual users are verified either with an external RADIUS server or with a local access database. XAuth is required for shared IKE IDs.

inet	Use an IPV4 address to identify the dynamic peer.
inet6	Use an IPV6 address to identify the dynamic peer.
reject-duplicate-connection	Reject new connection from duplicate IKE-id.
user-at-hostname	Use an e-mail address.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5. Support for the `inet6` option added in Junos OS Release 11.1.

`general-ikeid` option under `[edit security ike gateway` *`gateway-name`* `dynamic]` hierarchy is introduced in Junos OS Release 21.1R1.

RELATED DOCUMENTATION

IPsec Overview | 20

Example: Configuring AutoVPN with Pre-Shared Key | 1294

encryption-algorithm (Security IKE)

IN THIS SECTION

Syntax

```
encryption-algorithm (3des-cbc | aes-128-cbc | aes-128-gcm | aes-192-cbc | aes-256-cbc | aes-256-
gcm | des-cbc);
```

Hierarchy Level

```
[edit security ike proposal proposal-name]
```

Description

Configure an encryption algorithm for an IKE proposal. The device does not delete existing IPsec SAs when you update the `encryption-algorithm` configuration in the IKE proposal.

Options

`3des-cbc` Has a block size of 24 bytes; the key size is 192 bits long.

`aes-128-cbc` Advanced Encryption Standard (AES) 128-bit encryption algorithm.

`aes-128-gcm` AES 128-bit authenticated encryption algorithm supported with IKEv2 only. When this option is used, `aes-128-gcm` should be configured at the `[edit security ipsec proposal` *`proposal-name`*`]` hierarchy level, and the `authentication-algorithm` option should not be configured at the `[edit security ike proposal` *`proposal-name`*`]` hierarchy level.

When `aes-128-gcm` or `aes-256-gcm` encryption algorithms are configured in the IPsec proposal, it is not mandatory to configure AES-GCM encryption algorithm in the corresponding IKE proposal.

`aes-192-cbc` AES 192-bit encryption algorithm.

`aes-256-cbc` AES 256-bit encryption algorithm.

`aes-256-gcm` AES 256-bit authenticated encryption algorithm supported with IKEv2 only. When this option is used, `aes-256-gcm` should be configured at the `[edit security ipsec proposal` *`proposal-name`*`]` hierarchy level, and the `authentication-algorithm` option should not be configured at the `[edit security ike proposal` *`proposal-name`*`]` hierarchy level.

> **NOTE**: Integrity cannot be set with AES-GCM encryption algorithm.

`des-cbc` Has a block size of 8 bytes; the key size is 48 bits long.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5. Support for `aes-128-gcm` and `aes-256-gcm` options added in Junos OS Release 15.1X49-D40.

Starting in Junos OS Release 20.2R1, we've changed the help text description as `NOT RECOMMENDED` for the CLI options `3des-cbc` and `des-cbc` for devices running IKED with `junos-ike` package installed.

RELATED DOCUMENTATION

IPsec Overview | 20

proposal (Security IKE) | 1599

Installing Junos IKE package

encryption-failures

IN THIS SECTION

Syntax

```
encryption-failures {
    threshold value;
}
```

Hierarchy Level

```
[edit security alarms potential-violation]
```

Description

Raise a security alarm after exceeding a specified number of encryption failures.

Default

Multiple encryption failures do not cause an alarm to be raised.

Options

failures—Number of encryption failures up to which an alarm is not raised. When the configured number is exceeded, an alarm is raised.

- **Range:** 1 through 1,000,000,000.
- **Default:** 1000

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 11.2.

RELATED DOCUMENTATION

IPsec Overview | 20

Monitoring VPN Traffic | 1351

gateway (Security Group VPN Member IKE)

IN THIS SECTION

Syntax

```
gateway gateway-name {
    ike-policy policy-name;
    local address ip-address;
    local-identity {
        (hostname hostname | inet ip-address | inet6 ipv6-address | user-at-hostname e-mail-address);
    }
    remote-identity {
        (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
    }
    routing-instance routing-instance;
    server-address ip-address;
}
```

Hierarchy Level

```
[edit security group-vpn member ike]
```

Description

Configure IKE gateway for group VPN member. An IKE gateway initiates and terminates network connections between a firewall and a security device.

Options

gateway *gateway-name* Name of the gateway.

ike-policy *policy-name* Name of the IKE policy.

local address *ip-address* Configure the IPv4 address the member uses when accessing the group server.

local-identity *local-identity* Specify the local IKE identity to send in the exchange with the destination peer to establish communication.

remote-identity *remote-identity* Specify the name of a routing instance. If this is not specified, the default inet.0 routing instance is used.

routing-instance *routing-instance* Specify the name of a routing instance. If this is not specified, the default inet.0 routing instance is used.

server-address *ip-address* Specify the group server IPv4 address that this member registers through a `groupkey-pull` exchange. Up to four server IP addresses can be configured. The group member attempts to register with the first configured server. If registration with a configured server is not successful, the group member tries to register with the next configured server.

We recommend that group members only register with sub-servers in a server cluster and not the root-server.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2. Support for the `routing-instance` option added in Junos OS Release 15.1X49-D30 for vSRX Virtual Firewall.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

gateway (Security Group VPN Server IKE)

IN THIS SECTION

Syntax

```
gateway gateway-name {
    address ip-address;
    dead-peer-detection {
        always-send;
        interval seconds;
        threshold number;
    }
    dynamic {
        (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
    }
    ike-policy policy-name;
    local-address ip-address;
    local-identity {
        (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
    }
    remote-identity {
        (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
    }
    routing-instance routing-instance;
}
```

Hierarchy Level

```
[edit security group-vpn server ike]
```

Description

Configure IKE gateway for group VPN server.

Options

`gateway` *`gateway-name`* —Name of the gateway.

`address` *`ip-address`* —Specify the IP address of the peer.

`dead-peer-detection` —Enable DPD between group server cluster servers.

`dynamic`—Specify the identifier for the remote gateway with a dynamic IPv4 address. Use this statement to set up a VPN with a gateway that has an unspecified IPv4 address.

- `hostname` *`domain-name`* —Specify a fully qualified domain name.
- `inet` *`ip-address`* —Specify an IPv4 address to identify the dynamic peer.
- `user-at-hostname` *`e-mail-address`* —Specify an e-mail address.

Configuring `mode main` for group VPN servers or members is not supported when the remote gateway has a dynamic address and the authentication method is `pre-shared-keys.ike-policy` *`policy-name`* —Specify the name of the IKE policy.

`local-address` *`ip-address`* —Configure the source IP address the group VPN server uses when communicating with a group member or a root-server. This statement is normally used when there are multiple IP addresses bound to an interface.

`local-identity`—Specify the local IKE identity to send in the exchange with the destination peer to establish communication. If you do not configure a local-identity, the device uses the IPv4 corresponding to the local endpoint by default.

- `hostname` *`hostname`*—Specify identity as a fully qualified domain name (FQDN).
- `inet` *`ip-address`*—Specify identity as an IPv4 address.
- `user-at-hostname` *`e-mail-address`*—Specify identity as an e-mail address.

`remote-identity`—Specify the remote IKE identity of the destination peer. If you do not configure a remote identity, the device uses, by default, the IPv4 address that corresponds to the destination peer.

- `hostname` *`hostname`*—Specify identity as a fully qualified domain name (FQDN).
- `inet` *`ip-address`*—Specify identity as an IPv4 address.
- `user-at-hostname` *`e-mail-address`*—Specify identity as an e-mail address.

`routing-instance` *`routing-instance`*—Configure the routing instance that the group VPN server uses when communicating with a group server. This statement is used when the IKE gateway is not configured in the default routing instance.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

Support for the Group VPN server added in Junos OS Release 15.1X49-D30 for vSRX Virtual Firewall.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

gateway (Security IKE)

IN THIS SECTION

Syntax

```
gateway gateway-name {
    aaa {
        access-profile access-profile {
            config-payload-password config-payload-password;
        }
        client {
            password;
            username;
        }
    }
    address [ip-address-or-hostname];
    advpn {
        suggester {
            disable;
        }
        partner {
            connection-limit number;
            idle-threshold   packets/sec;
            idle-time        seconds;
            disable;
        }
    }
    dead-peer-detection {
        (always-send | optimized | probe-idle-tunnel);
        interval seconds;
        threshold number;
    }
    dynamic {
        connections-limit number;
        distinguished-name {
            container container-string;
            wildcard wildcard-string
        }
        general-ikeid;
        hostname domain-name;
        ike-user-type (group-ike-id | shared-ike-id);
        inet ip-address;
        inet6 ipv6-address;
        reject-duplicate-connection;
        user-at-hostname e-mail-address;
    }
    external-interface external-interface-name;
    fragmentation {
        disable;
        size bytes;
    }
    general-ikeid;
    ike-policy policy-name;
    local-address (ipv4-address | ipv6-address);
    local-identity {
        (distinguished-name | hostname hostname | inet ip-address | inet6 ipv6-address | key-id
| user-at-hostname e-mail-address);
```

```
    }
    nat-keepalive seconds;
    no-nat-traversal;
    node-local;
    remote-identity {
        (distinguished-name <container container-string> <wildcard wildcard-string> | hostname
hostname | inet ip-address | inet6 ipv6-address | key-id | user-at-hostname e-mail-address);
    }
    tcp-encap-profile profile-name;
    version (v1-only | v2-only);
}
```

Hierarchy Level

```
[edit security ike]
```

Description

Configure an IKE gateway.

Options

gateway-name Name of the gateway.

address Specify the IPv4 or IPv6 address or the hostname of the primary Internet Key Exchange (IKE) gateway and up to four backup gateways.

- Values:
  - address—IPv4 or IPv6 address or hostname of an IKE gateway.

aaa Specify that extended authentication is performed in addition to IKE Phase 1 authentication for remote users trying to access a VPN tunnel.

advpn

Enable Auto Discovery VPN (ADVPN) protocol on the specified gateway. ADVPN dynamically establishes VPN tunnels between spokes to avoid routing traffic through the Hub.

dead-peer-detection

Enable the device to use dead peer detection (DPD).

dynamic

Specify the identifier for the remote gateway with a dynamic IPv4 or IPv6 address. Use this statement to set up a VPN with a gateway that has an unspecified IPv4 or IPv6 address.

external-interface

Name of the interface to be used to send traffic to the IPsec VPN. Specify the outgoing interface for IKE SAs. This interface is associated with a zone that acts as its carrier, providing firewall security for it.

fragmentation

Disable IKEv2 packet fragmentation and, optionally, configure the maximum size of an IKEv2 message before the message is split into fragments that are individually encrypted and authenticated.

disable

Disables IKEv2 fragmentation. IKEv2 fragmentation is enabled by default.

size *bytes*

Maximum size, in bytes, of an IKEv2 message before it is split into fragments. The size applies to both IPv4 and IPv6 messages.

- **Range:** 500 to 1300 bytes
- **Default:** 576 bytes for IPv4 messages and 1280 bytes for IPv6 messages

general-ikeid

Accept peer IKE-ID in general.

ike-policy

Specify the IKE policy to be used for the gateway.

local-address

Local IP address for IKE negotiations. Specify the local gateway address. Multiple addresses in the same address family can be configured on an external physical interface to a VPN peer. If this is the case, we recommend that `local-address` be configured. If there is only one address configured (IPv4 or IPv6) on an external physical interface, `local-address` configuration is not necessary.

The `local-address` value must be an IP address that is configured on an interface on the SRX Series Firewall. We recommend that `local-address` belong to the external interface of the IKE gateway. If local-address does not belong to the external interface of the IKE gateway, the interface must be in the same zone as the external interface of the IKE gateway and an intra-zone security policy must be configured to permit traffic. The `local-address` value and the remote IKE gateway address must be in the same address family, either IPv4 or IPv6.

local-identity
Specify the local IKE identity to send in the exchange with the destination peer to establish communication.

nat-keepalive
Specify the interval at which NAT keepalive packets (seconds) can be sent so that NAT translation continues. Default value changed from 5 seconds to 20 seconds in Junos OS Release 12.1X46-D10.

- **Default:** 20
- **Range:** 1 through 300

node-local
Mark an IPsec VPN tunnel between Multinode High Availability nodes and a VPN peer device as a node-local tunnel. Node-local tunnels support dynamic routing protocols that facilitate the device to add the routes dynamically. These routes remain local to a node and are not bound to any services redundancy group (SRG). Use this option only for Multinode High Availability.

no-nat-traversal
Disable IPSec NAT traversal. Disables UDP encapsulation of IPsec Encapsulating Security Payload (ESP) packets, otherwise known as Network Address Translation Traversal (NAT-T). NAT-T is enabled by default.

tcp-encap-profile
Specify the TCP encapsulation profile to be used for TCP connections for remote access clients.

version
Specify the IKE version to use to initiate the connection.

- Values:
  - v1-only—The connection must be initiated using IKE version 1. This is the default.
  - v2-only—The connection must be initiated using IKE version 2

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1. The `inet6` option added in Junos OS Release 11.1. Support for the `advpn` option added in Junos OS Release 12.3X48-D10.

Option `fragmentation` is introduced in Junos OS Release 15.1X49-D80.

Option `node-local` is introduced in Junos OS Release 23.2R1.

`general-ikeid` option under `[edit security ike gateway` *`gateway-name`* `dynamic]` hierarchy is introduced in Junos OS Release 21.1R1.

RELATED DOCUMENTATION

IPsec Overview | **20**

Example: Configuring AutoVPN with Pre-Shared Key | **1294**

general-ikeid

IN THIS SECTION

Syntax

```
general-ikeid;
```

Hierarchy Level

```
[set security ike gateway gateway_name dynamic]
```

Description

During IKE Phase 1 negotiation, when negotiation request is received, there are two identity checks.

1. IKE-ID validation from ID payload.

2. Phase 1 authentication by pre-shared key or RSA/DSA certificate.

Configure `remote-identity` to lookup the certificate of the peer for certificate authentication. This `remote-identity` should match the corresponding field in the `SubjectAltname` extension of the peer certificate for successful detection of peer certificate and authentication.

The identity check with the same IKE-ID is repeated, that is, the IKE-ID validation with remote-identity and the certificate authentication. To avoid this, during authentication of remote peer, use the `general-ikeid` under the`set security ike gateway *gateway_name* dynamic` hierarchy level to bypass the validation process.

If you enable this option, then during authentication of remote peer, the device accepts all ike-id types like, hostname, user@hostname, and so on.

Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 21.1R1

RELATED DOCUMENTATION

Example: Configuring AutoVPN with Pre-Shared Key

global-options (Juniper Secure Connect)

IN THIS SECTION

Syntax

```
global-options {
    auth-token-valid-time seconds;
}
```

Hierarchy Level

```
[edit security remote-access]
```

Description

Define global parameters for Juniper Secure Connect remote access configuration.

Options

auth-token-valid-time

Authentication token valid time (seconds).

- **Default:** 60 seconds
- **Range:** 1 through 300

Required Privilege Level

security

Release Information

Statement introduced in Junos OS Release 20.3R1

RELATED DOCUMENTATION

Juniper Secure Connect Administrator Guide

group (Security Group VPN)

IN THIS SECTION

Syntax

```
group name {
    anti-replay-time-window milliseconds;
    description description;
    group-id number;
    ike-gateway gateway-name;
    ipsec-sa name {
        match-policy policy-name {
            destination ip-address/netmask;
            destination-port number;
            protocol number;
            source ip-address/netmask;
            source-port number;
        }
        proposal proposal-name;
    }
    member-threshold number;
    server-cluster {
        ike-gateway gateway-name;
        retransmission-period seconds;
        server-role (root-server | sub-server);
    }
    server-member-communication {
        certificate certificate-id;
        communication-type (unicast);
        encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
        lifetime-seconds seconds;
        number-of-retransmission number;
        retransmission-period seconds;
        sig-hash-algorithm (sha-256 | sha-384);
    }
}
```

Hierarchy Level

```
[edit security group-vpn server]
```

Description

Configure group VPN on the group server. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

Options

name—Name of the group.

- `anti-replay-time-window` *milliseconds*—Configure antireplay time in milliseconds. Specify a value from 1 to 60,000.

  We recommend that NTP be configured on Group VPNv2 devices to ensure proper antireplay operation.

  Group members that are running on vSRX Virtual Firewall instances on a host machine where the hypervisor is running under a heavy load may experience issues that can be corrected by reconfiguring the `anti-replay-time-window` value. If data that matches the IPsec policy on the group member is not being transferred, check the `show security group-vpn member ipsec statistics` output for D3P errors. Make sure that NTP is operating correctly. If there are errors, adjust the `anti-replay-time-window` value.

- `description` *description*—Description of the group.

- `group-id` *number*—Identifier for this group VPN. Specify a value from 1 to 4,294,967,295.

- `ike-gateway` *gateway-name*—Define the group member for Phase 1 negotiation. There can be multiple instances of this option configured. When a group member sends its registration request to the server, the server checks to see that the member is configured for the group.

- `ipsec-sa` *name*—Configure the group SAs to be downloaded to members. There can be multiple group SAs downloaded to group members.

- `member-threshold` *number*—Specify the maximum number of group VPN members that can be accepted in the group. The same `member-threshold` value must be configured on the root-server and all sub-servers in a group server cluster.

  The maximum number you can configure for a group is dependent upon the group server platform. Also, the sum of the `member-threshold` numbers for all groups configured on the group server must not exceed the capacity of the group server platform.

- server-cluster—Configure the Group Domain of Interpretation (GDOI) group controller/key server (GCKS) cluster for the specified group. All servers in a group VPN server cluster must be SRX Series Firewalls.
- server-member-communication—Enable and configure server to member communication. When these options are configured, group members receive new keys before current keys expire.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2

member-threshold option introduced in Junos OS Release 15.1X49-D30 for vSRX Virtual Firewall.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

group-vpn

IN THIS SECTION

Syntax

```
group-vpn {
    member {
        ike {
            gateway gateway-name;
                policy;
                proposal;
                traceoptions;
            }
        ipsec {
            vpn vpn-name {
                df-bit (clear | copy | set);
                exclude rule rule-name {
                    source-address ip-address/mask;
                    destination-address ip-address/mask;
                    application application;
                }
                fail-open rule rule-name {
                    source-address ip-address/mask;
                    destination-address ip-address/mask;
                    application application;
                }
                group id;
                group-vpn-external-interface interface;
                ike-gateway gateway-name;
                recovery-probe;
            }
        }
    }
    server {
        group name {
            anti-replay-time-window milliseconds;
            description description;
            group-id number;
            ike-gateway gateway-name;
            ipsec-sa;
            member-threshold number;
            server-cluster;
        }
        ike {
            gateway gateway-name;
                policy;
            proposal;
        }
        ipsec {
            proposal proposal-name;
        }
        traceoptions (Security Group VPN);
    }
}
```

Hierarchy Level

```
[edit security]
```

Description

Configure Group VPNs in Group VPNv2. Group VPNv2 extends IPsec architecture to support SAs that are shared by a group of security devices. With Group VPNv2, any-to-any connectivity is achieved by preserving the original source and destination IP addresses in the outer header.

Options

member Configure group VPN member.

ike Configure IPsec group VPN on the group member.

policy Configure an IKE policy.

proposal

Define an IKE proposal. You can configure one or more IKE proposals. Each proposal is a list of IKE attributes to protect the IKE connection between the IKE host and its peer.

traceoptions

Configure group VPN tracing options to aid in troubleshooting the IKE or server issues.

ipsec

Configure IPsec for Phase 2 exchange on the group member.

vpn

Configure IPsec VPN for Phase 2 exchange on the group member.

server

Configure group VPN server.

group

Configure group VPN on the group server.

anti-replay-time-window

Configure antireplay time in milliseconds. Specify a value from 1 to 60,000. Each IPsec packet contains a timestamp. The group member checks whether the packet's timestamp falls within the configured `anti-replay-time-window` value. A packet is dropped if the timestamp exceeds the value.

description

Description of the group.

group-id ***number***

Identifier for this group VPN. Specify a value from 1 to 4,294,967,295.

ike-gateway ***gateway-name***

Define the group member for Phase 1 negotiation. There can be multiple instances of this option configured. When a group member sends its registration request to the server, the server checks to see that the member is configured for the group.

ipsec-sa

Configure the group SAs to be downloaded to members. There can be multiple group SAs downloaded to group members.

member-threshold

Specify the maximum number of group VPN members that can be accepted in the group. There is no default number.

server-cluster

Configure the Group Domain of Interpretation (GDOI) group controller/key server (GCKS) cluster for the specified group. All servers in a group VPN server cluster must be SRX Series Firewalls.

server-member-communication

Enable and configure server to member communication. When these options are configured, group members receive new keys before current keys expire.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

ike (High Availability)

IN THIS SECTION

Syntax

```
ike {
    gateway name {
        ike-policy policy-name;
        version (v1-only | v2-only);
    }
    policy name {
        description description;
        pre-shared-key (ascii-text ascii-text | hexadecimal hexadecimal);
        proposals [ proposals ... ];
    }
    proposal name {
        authentication-algorithm (md5 | sha-256 | sha-384 | sha-512 | sha1);
        authentication-method (dsa-signatures | ecdsa-signatures-256 | ecdsa-signatures-384 |
ecdsa-signatures-521 | pre-shared-keys | rsa-signatures);
        description description;
        dh-group (group1 | group14 | group15 | group16 | group19 | group2 | group20 | group21 |
group24 | group5);
        encryption-algorithm (aes-256-gcm);
        lifetime-seconds seconds;
    }
}
```

Hierarchy Level

```
[edit security]
```

Description

Define Internet Key Exchange (IKE) configuration for high availability feature. IKE is a key management protocol that creates dynamic SAs; it negotiates SAs for IPsec. An IKE configuration defines the algorithms and keys used to establish a secure connection with a peer security gateway.

Options

gateway-name Name of the gateway.

ike-policy Specify the IKE policy to be used for the gateway.

version

Specify the IKE version to use to initiate the connection.

- Values:
  - v2-only—The connection must be initiated using IKE version 2. For Multinode High availability feature, you must configure the IKE version as `v2-only`.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 20.4R1.

RELATED DOCUMENTATION

ipsec (High Availability) | **1544**

High-Availability (Chassis)

Multinode High Availability

services-redundancy-group

local-id

peer-id

traceoptions

show security ike security-associations | **1851**

show security ike active-peer | **1838**

show security ipsec security-associations | **1899**

show security ipsec statistics | **1938**

clear security ike security-associations | **1696**

clear security ipsec security-associations | **1698**

ike (Security)

IN THIS SECTION

Syntax

```
ike {
    gateway (Security IKE) name {
        ( address | dynamic (Security) distinguished-name (Security) < container> < wildcard>
hostname inet inet6  user-at-hostname <connections-limit connections-limit> <ike-user-type
(group-ike-id | shared-ike-id)> <reject-duplicate-connection>);
        aaa {
             access-profile;
            client password password username username;
        }
        advpn {
            partner {
                connection-limit connection-limit;
                disable;
                idle-threshold idle-threshold;
                idle-time seconds;
            }
            suggester {
                disable;
            }
        }
        dead-peer-detection (always-send | optimized | probe-idle-tunnel);
        external-interface external-interface;
        fragmentation {
            disable;
            size size;
        }
        general-ikeid;
         ike-policy;
         local-address;
        local-identity (distinguished-name | hostname identity-hostname | inet identity-ipv4 |
inet6 identity-ipv6 | key-id string-key-id | user-at-hostname identity-user);
        remote-identity distinguished-name <container container> <wildcard wildcard>hostname
identity-hostnameinet identity-ipv4inet6 identity-ipv6 key-id string-key-id user-at-hostname
identity-user;
         tcp-encap-profile;
        version (v1-only | v2-only);
    }
    policy name {
        certificate {
            local-certificate (Security) local-certificate;
            peer-certificate-type (pkcs7 | x509-signature);
            policy-oids policy-oids;
            trusted-ca (ca-profile ca-profile | trusted-ca-group trusted-ca-group  );
        }
        description description;
        mode (aggressive | main);
        pre-shared-key (ascii-text ascii-text | hexadecimal hexadecimal);
        seeded-pre-shared-key (ascii-text key | hexadecimal key);
        proposal-set (Security IKE) (basic | compatible | prime-128 | prime-256 | standard | suiteb-
gcm-128 | suiteb-gcm-256);
        proposals [ proposals ... ];
        reauth-frequency reauth-frequency;
    }
    proposal proposal-name {
        authentication-algorithm (md5 | sha-256 | sha-384 | sha-512 | sha1);
        authentication-method (certificates | dsa-signatures | ecdsa-signatures-256 | ecdsa-
signatures-384 | ecdsa-signatures-521 | pre-shared-keys | rsa-signatures);
        description description;
        dh-group dh-group (group1 | group14 | group15 | group16 | group19 | group2 | group20 |
group21 | group24 | group5);
        encryption-algorithm  (3des-cbc | aes-128-cbc | aes-128-gcm | aes-192-cbc | aes-256-cbc |
aes-256-gcm | des-cbc);
        lifetime-seconds seconds;
    }
    respond-bad-spi <max-responses>;
    traceoptions {
        file {
            filename;
            files number;
            match regular-expression;
            size maximum-file-size;
            (world-readable | no-world-readable);
        }

        level (critical | error | terse | warning | detail);
        flag flag (all | certificates | config | database | general | high-availability | ike |
next-hop-tunnels | parse | policy-manager | routing-socket | thread | timer)reference/
configuration-statement/security-edit-ike-security;
        no-remote-trace;
        rate-limit messages-per-second;
    }
}
```

Hierarchy Level

```
[edit security]
```

Description

Define Internet Key Exchange (IKE) configuration. IKE is a key management protocol that creates dynamic SAs; it negotiates SAs for IPsec. An IKE configuration defines the algorithms and keys used to establish a secure connection with a peer security gateway.

Options

`respond-bad-spi` *max-responses*—(Optional) Number of times to respond to invalid SPI values per gateway. Enable response to invalid IPsec Security Parameter Index (SPI) values. If the security associations (SAs) between two peers of an IPsec VPN become unsynchronized, the device resets the state of a peer so that the two peers are synchronized.

- **Range:** 1 through 30
- **Default:** 5

`traceoptions`—Configure IKE tracing options to aid in troubleshooting the IKE issues. This helps troubleshoot one or multiple tunnels negotiation by standard tracefile configuration. IKE tracing allows the user to view the detailed packet exchange and the negotiation information in Phase 1 and Phase 2. IKE tracing is not enabled by default. By default , all IKE or IPsec negotiations are logged into /var/log/kmd. But user can also specify customized file name while configuring the IKE traceoptions.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5.

Support for IPv6 addresses added in Junos OS Release 11.1.

Support for `inet6` option added in Junos OS Release 11.1.

Support for `group15`, `group16`, `group21`, `ecdsa-signatures-521`, and `sha-512` options added in Junos OS Release 19.1R1 on SRX5000 line with `junos-ike` package installed.

Starting in Junos OS Release 20.2R1, we've changed the help text description as `NOT RECOMMENDED` for the CLI options `md5` and `sha1` for devices running IKED with `junos-ike` package installed.

Support for `group15`, `group16`, and `group21` options added in Junos OS Release 20.3R1 on vSRX Virtual Firewall instances with `junos-ike` package installed.

Support for `group15`, `group16`, and `group21` options added in Junos OS Release 21.1R1 on vSRX Virtual Firewall 3.0 instances with `junos-ike` package installed.

`level` option introduced in Junos OS Release 21.1R1.

Support for `seeded-pre-shared-key` option added in Junos OS Release 21.1R1.

RELATED DOCUMENTATION

IPsec Overview | 20

ALG Overview

Understanding Logical Systems for SRX Series Services Gateways

Example: Configuring AutoVPN with Pre-Shared Key | 1294

Installing Junos IKE package

ike (Security Group VPN Member)

IN THIS SECTION

Syntax

```
ike {
    gateway gateway-name {
        ike-policy policy-name;
        local address ip-address;
        local-identity {
            (hostname hostname | inet ip-address | inet6 ipv6-address | user-at-hostname e-mail-
address);
        }
        remote-identity {
            (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
        }
        routing-instance routing-instance;
        server-address ip-address;
    }
    policy policy-name {
        description description;
        mode (aggressive | main);
        pre-shared-key (ascii-text key | hexadecimal key);
        proposals proposal-name;
    }
    proposal proposal-name {
        authentication-algorithm (sha-256 | sha-384);
        authentication-method pre-shared-keys;
        description description;
        dh-group (group14 | group24);
        encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
        lifetime-seconds seconds;
    }
    traceoptions {
        file {
            filename;
            files number;
            match regular-expression;
            size maximum-file-size;
            (world-readable | no-world-readable);
        }
        flag flag (all | certificates | config | database | general | high-availability | ike |
next-hop-tunnels | parse | policy-manager | routing-socket | thread | timer);
        gateway-filter {
            local-address ip-address;
            remote-address ip-address;
        }
        level (all | error | info | notice | verbose | warning);
        no-remote-trace;
    }
}
```

Hierarchy Level

```
[edit security group-vpn member]
```

Description

Configure IKE group VPN on the group member. A group member encrypts the traffic and is responsible for the actual encryption and decryption of data traffic. A group member is configured with IKE Phase 1 parameters and GC/KS information.

Options

gateway *gateway-name* Configure IKE gateway for group VPN member.

policy *policy-name* Configure an IKE policy.

proposal*proposal-name* Define an IKE proposal.

traceoptions Configure group VPN tracing options to aid in troubleshooting the IKE issues.

ipsec Configure IPsec for Phase 2 exchange on the group member.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | **758**

ike (Security Group VPN Server)

IN THIS SECTION

Syntax

```
ike {
    gateway gateway-name {
        address ip-address;
        dead-peer-detection {
            always-send;
            interval seconds;
            threshold number;
        }
        dynamic {
            (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
        }
        ike-policy policy-name;
        local-address ip-address;
        local-identity {
            (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
        }
        remote-identity {
            (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
        }
        routing-instance routing-instance;
    }
    policy policy-name {
        description description;
        mode (aggressive | main);
        pre-shared-key (ascii-text key | hexadecimal key);
        proposals proposal-name;
    }
    proposal proposal-name {
        authentication-algorithm (sha-256 | sha-384);
        authentication-method pre-shared-keys;
        description description;
        dh-group (group14 | group24);
        encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
    }
}
```

Hierarchy Level

```
[edit security group-vpn server]
```

Description

Configure Phase 1 security association (SA) with a member on the group server. The gateway is the group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

Options

gateway *gateway-name*	Configure IKE gateway for group VPN server.
policy *policy-name*	Configure an IKE policy.
proposal *proposal-name*	Define an IKE proposal.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

ike (Security IPsec VPN)

IN THIS SECTION

Syntax

```
ike {
    anti-replay-window-size anti-replay-window-size;
    gateway gateway-name;
    idle-time seconds;
    install-interval seconds;
    ipsec-policy ipsec-policy-name;
    no-anti-replay;
    proxy-identity {
        local ip-prefix;
        remote ip-prefix;
        service (any | service-name);
    }
}
```

Hierarchy Level

```
[edit security ipsec vpn vpn-name]
```

Description

Define an IKE-keyed IPsec VPN.

Options

anti-replay-window-size

To enable the `anti-replay-window-size` option, you first need to configure the option for each VPN object or at the global level. You can configure the anti-replay window size in the range of 64 to 8192 (power of 2). If the anti-replay window size is not configured, the window size is 64 by default. If `anti-replay-window-size` command is configured at both the global and VPN object levels, the configuration on VPN object takes precedence over global configuration.

`anti-replay-window-size` is supported only on SRX5000 line with SRX5K-SPC3 card installed.

gateway-name

Name of the remote IKE gateway.

idle-time

Specify the maximum amount of idle time to delete a security association (SA).

- **Default:** To be disabled
- **Range:** 60 through 999,999 seconds

install-interval

Specify the maximum number of seconds to allow for the installation of a rekeyed outbound security association (SA) on the device.

- **Default:** 1 second
- **Range:** 0 through 10 seconds

ipsec-policy

Specify the IPsec policy name.

no-anti-replay

Disable the antireplay checking feature of IPsec. Antireplay is an IPsec feature that can detect when a packet is intercepted and then replayed by attackers. By default, antireplay checking is enabled.

proxy-identity

Optionally specify the IPsec proxy ID to use in negotiations. The default is the identity based on the IKE gateway. If the IKE gateway is an IPv6 site-to-site gateway, the default proxy ID is ::/0. If the IKE gateway is an IPv4 gateway or a dynamic endpoint or dialup gateway, the default proxy ID is 0.0.0.0/0.

- `local`—Specify the local IPv4 or IPv6 address and subnet mask for the proxy identity.
- `remote`—Specify the remote IPv4 or IPv6 address and subnet mask for the proxy identity.
- `service`—Specify the service (port and protocol combination) to protect. Name of the service is as defined with `system-services` (`Interface Host-Inbound Traffic`) and `system-services` (`Zone Host-Inbound Traffic`).

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5. Support.

Statement `anti-replay-window-size` is introduced in Junos OS Release 19.2R1.

RELATED DOCUMENTATION

IPsec Overview | 20

ike-phase1-failures

IN THIS SECTION

Syntax

```
ike-phase1-failures {
    threshold value;
}
```

Hierarchy Level

```
[edit security alarms potential-violation]
```

Description

Raise a security alarm after exceeding a specified number of Internet Key Exchange (IKE) Phase 1 failures.

Default

Multiple IKE phase 1 failures do not cause an alarm to be raised.

Options

failures—Number of IKE phase 1 failures up to which an alarm is not raised. When the configured number is exceeded, an alarm is raised.

- **Range:** 1 through 1,000,000,000.
- **Default:** 20

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 11.2.

RELATED DOCUMENTATION

IPsec Overview | 20

Monitoring VPN Traffic | 1351

ike-phase2-failures

IN THIS SECTION

Syntax

```
ike-phase2-failures {
    threshold value;
}
```

Hierarchy Level

```
[edit security alarms potential-violation]
```

Description

Raise a security alarm after exceeding a specified number of Internet Key Exchange (IKE) phase 2 failures.

Default

Multiple IKE phase 2 failures do not cause an alarm to be raised.

Options

`failures`—Number of IKE phase 2 failures up to which an alarm is not raised. When the configured number is exceeded, an alarm is raised.

- **Range:** 1 through 1,000,000,000.
- **Default:** 20

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 11.2.

RELATED DOCUMENTATION

IPsec Overview | 20

Monitoring VPN Traffic | 1351

inline-fpga-crypto

IN THIS SECTION

Syntax

```
inline-fpga-crypto (disabled | enabled);
```

Hierarchy Level

```
[edit security forwarding-process application-services (Security Forwarding Process)]
```

Description

Enable or disable Inline FPGA Decryption (IFD) for cryptographic operation.

By default, the inline-fpga-crypto is disabled.

PowerMode IPsec (PMI) uses AES-NI and FPGA for decryption of cryptographic operation to enhance IPSec VPN performance.

PMI with FPGA is supported on SRX5400, SRX5600, and SRX5800 with SPC3 cards.

Options

disabled	Disable inline FPGA crypto
enabled	Enable inline FPGA crypto

Required Privilege Level

security

Release Information

Statement introduced in Junos OS Release 20.4R1.

RELATED DOCUMENTATION

Improving IPsec Performance with PowerMode IPsec | 1383

application-services (Security Forwarding Process)

internal (Security IPsec)

IN THIS SECTION

Syntax

```
internal {
    security-association {
        manual {
            encryption {
                algorithm (3des-cbc | aes-128-cbc);
                ike-ha-link-encryption enable;
                key ascii-text;
            }
        }
    }
}
```

Hierarchy Level

```
[edit security ipsec]
```

Description

Enable secure login and to prevent attackers from gaining privileged access through this control port by configuring the internal IP security (IPsec) security association (SA).

When the internal IPsec is configured, IPsec-based `rlogin` and remote command (`rcmd`) are enforced, so an attacker cannot gain unauthorized information.

Options

security-association — Specify an IPsec SA. An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec.

manual encryption — Specify a manual SA. Manual SAs require no negotiation; all values, including the keys, are static and specified in the configuration.

algorithm 3des-cbc

Specify the encryption algorithm for the internal Routing-Engine-to-Routing-Engine IPsec SA configuration.

algorithm aes-128-cbc

Specify the encryption algorithm for high availability encryption link.

iked-ha-link-encryption

Enable encryption for internal messages.

- Values:
  - `enable`—Enable HA link encryption IKE internal messages

key ascii-text

Specify the encryption key. You must ensure that the manual encryption key is in ASCII text and 24 characters long; otherwise, the configuration will result in a commit failure.

Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.1X45-D10.

Support for `ike-ha-link-encryption` option added in Junos OS Release 12.1X47-D15.

Support for iked_encryption option added in Junos OS Release 12.1X47-D10.

Support for `aes-128-cbc` option added in Junos OS Release 19.1R1.

Support for `ike-ha-link-encryption` option added for vSRX Virtual Firewall in Junos OS Release 19.4R1

RELATED DOCUMENTATION

Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Devices

show security internal-security-association

ipsec (High Availability)

IN THIS SECTION

Syntax

```
ipsec {
    vpn vpn-name {
        ha-link-encryption;
        ike {
            gateway gateway-name;
            ipsec-policy ipsec-policy-name;
        }
    }
    proposal proposal-name {
        description description;
        encryption-algorithm (aes-256-gcm);
        lifetime-seconds seconds;
        protocol (esp);
    }
    policy policy-name {
        description description;
        proposals proposal-name;
    }
}
```

Hierarchy Level

```
[edit security]
```

Description

Define IPsec configuration for the multinode high availability feature. A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. An IPsec tunnel is created between two participant devices to secure VPN communication.

Options

vpn-name Configure an IPsec VPN. A VPN provides a means by which remote computers communicate securely across a public WAN such as the Internet.

You must mention the same VPN name for `vpn-profile` in `set chassis high-availability peer-id` *`peer-id`* `vpn-profile` *`profile-name`* configuration.

ha-link-encryption Configure a interchassis link tunnel for secure HA traffic flow between the nodes. Only site-to-site IPsec VPN tunnels are supported for interchassis link tunnels. Both PSK and PKI authentication methods are supported.

gateway-name Name of the remote IKE gateway.

ipsec-policy-name Specify the IPsec policy name.

proposal-name Name of the IPsec proposal. An IPsec proposal lists protocols and algorithms (security services) to be negotiated with the remote IPsec peer.

description Text description of IPsec proposal.

encryption-algorithm Define encryption algorithm. The device deletes existing IPsec SAs when you update the `encryption-algorithm` configuration in the IPsec proposal.

A commit error is thrown if any value other than `aes-256-gcm` is configured.

- Values:
  - `aes-256-gcm`—AES GCM 256-bit encryption algorithm.

    For an IKE proposal, AES 256-bit authenticated encryption algorithm is supported with IKEv2 only. When this option is used, `aes-256-gcm` should be configured at the [`edit security ipsec proposal` *`proposal-name`*] hierarchy level, and the `authentication-algorithm` option should not be configured at the [`edit security ike proposal` *`proposal-name`*] hierarchy level.

lifetime-seconds

Lifetime in seconds.

- **Range:** 180 through 86400
- **Default:** 3600 seconds

protocol

Define the IPsec protocol for a manual or dynamic security association (SA).

A commit error is thrown if any value other than `esp` is configured.

- Values:
  - esp—Encapsulated Security Payload header

policy-name

Define an IPsec policy. An IPsec policy defines a combination of security parameters (IPsec proposals) used during IPsec negotiation. It defines Perfect Forward Secrecy (PFS) and the proposals needed for the connection.

description

Enter descriptive text for an IPsec policy.

proposal-name

Specify one or more proposals for an IPsec policy.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 20.4R1.

RELATED DOCUMENTATION

- ike (High Availability) | 1521
- High-Availability (Chassis)
- Multinode High Availability
- services-redundancy-group
- local-id
- peer-id
- traceoptions
- show security ike security-associations | 1851
- show security ike active-peer | 1838
- show security ipsec security-associations | 1899
- show security ipsec statistics | 1938
- clear security ike security-associations | 1696
- clear security ipsec security-associations | 1698

ipsec (Security)

IN THIS SECTION

Syntax

```
ipsec {
    anti-replay-window-size anti-replay-window-size;
    internal;
    policy;
    proposal
    security-association sa-name;
    traceoptions;
    vpnvpn-name;
    vpn-monitor-options {
        interval seconds;
        threshold number;
    }
}
```

Hierarchy Level

```
[edit security]
```

Description

Define IPsec configuration. A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. An IPsec tunnel is created between two participant devices to secure VPN communication.

Options

anti-replay-window-size

Anti-replay window size.

- **Range:** 64 through 8192 bytes
- **Default:** 64 bytes

internal

Configure internal IPsec. When the internal IPsec is configured, IPsec-based `rlogin` and remote command (`rcmd`) are enforced, so an attacker cannot gain unauthorized information.

policy

Define an IPsec policy. An IPsec policy defines a combination of security parameters (IPsec proposals) used during IPsec negotiation. It defines Perfect Forward Secrecy (PFS) and the proposals needed for the connection.

proposal

Name of the IPsec proposal. An IPsec proposal lists protocols and algorithms (security services) to be negotiated with the remote IPsec peer.

security-association

Configure a manual IPsec security association (SA) to be applied to an OSPF or OSPFv3 interface or virtual link. IPsec can provide authentication and confidentiality to OSPF or OSPFv3 routing packets.

traceoptions

Configure IPsec tracing options. Trace operations track IPsec events and record them in a log file in the /var/log directory.

vpn *vpn-name*

Configure an IPsec VPN. A VPN provides a means by which remote computers communicate securely across a public WAN suchas the Internet

vpn-monitor-options

Configure VPN monitoring options

interval *seconds*

Interval at which to send ICMP requests to the peer.

- **Range:** 2 through 3600 seconds
- **Default:** 10 seconds

threshold *number*

Number of consecutive unsuccessful pings before the peer is declared unreachable.

- **Range:** 1 through 65,536 pings
- **Default:** 10 pings

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5.

`group15`, `group16`, `group21`, `hmac-sha-512` and `hmac-sha-384` options introduced in Junos OS Release 19.1R1 on SRX Series Firewalls.

RELATED DOCUMENTATION

IPsec Overview | 20

ipsec (Security Group VPN Member)

IN THIS SECTION

Syntax

```
ipsec {
    vpn vpn-name {
        df-bit (clear | copy | set);
        exclude rule rule-name {
            source-address ip-address/mask;
            destination-address ip-address/mask;
            application application;
        }
        fail-open rule rule-name {
            source-address ip-address/mask;
            destination-address ip-address/mask;
            application application;
        }
        group id;
        group-vpn-external-interface interface;
        ike-gateway gateway-name;
        recovery-probe;
    }
t}
```

Hierarchy Level

```
[edit security group-vpn member]
```

Description

Configure IPsec for Phase 2 exchange on the group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

Options

vpn *vpn-name*

Name of the VPN.

df-bit

Specifies pre-fragmentation and post-fragmentation of IPsec traffic on the group member. One of the following options can be configured:

- clear—Sets the outer IP do not fragment (DF) bit to 0. When the packet size is larger than the path maximum transmission unit (path MTU), pre-fragmentation is done if the DF bit is not set in the inner packet and post-fragmentation is done if the DF bit is set in the inner packet. This is the default.

- copy—Copies the DF bit from the inner header to the outer header. When the packet size is larger than the path PMTU, pre-fragmentation is done if the DF bit is not set in the inner packet. If the DF bit is set in the inner packet, the packet is dropped and an ICMP message is sent back.

- set—Sets the outer IP DF bit to 1. When the packet size is larger than the path MTU, pre-fragmentation is done if the DF bit is not set in the inner packet. If the DF bit is set in the inner packet, the packet is dropped and an ICMP message is sent back

exclude rule

Specifies traffic to be excluded from Group VPN encryption. A maximum of 10 exclude rules can be configured. Source and destination addresses must be specified in *ip-address/ mask* format; address books and address sets are not supported. Predefined and user-defined applications are supported, but application sets are not supported.

fail-open rule

Specifies the traffic to be sent in cleartext mode if there is no valid SA key available to protect the traffic. Traffic that is not specified by the fail-open rule is blocked if there is no valid SA key available to protect the traffic. A maximum of 10 fail-open rules can be configured. Source and destination addresses must be specified in *ip-address/ mask* format; address books and address sets are not supported. Predefined and user-defined applications are supported, but application sets are not supported.

group *id*

Identifier configured for the Group VPN.

group-vpn-external-interface *interface*

Interface used by the group member to connect to the Group VPN peers. The interface must belong to the same zone as the `to-zone` configured at the [`edit security ipsec-policy`] hierarchy level for Group VPN traffic.

ike-gateway *gateway-name*

Name of the IKE gateway for the Group VPN.

recovery-probe

Enables initiation of `groupkey-pull` exchanges at specific intervals to update the member's SA from the group server if the group member is determined to be out of synchronization with the group server and other group members. This option is disabled by default.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2. `df-bit`, `exclude rule`, `fail-open rule`, and `recovery-probe` options added in Junos OS Release 15.1X49-D30 for vSRX Virtual Firewall.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

ipsec (Security Group VPN Server)

IN THIS SECTION

Syntax

```
ipsec {
    proposal proposal-name {
        authentication-algorithm (hmac-sha-256-128);
        description description;
        encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
        lifetime-seconds seconds;
    }
}
```

Hierarchy Level

```
[edit security group-vpn server]
```

Description

Configure IPsec proposal for Phase 2 exchange on the group server. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

Options

`proposal` *`proposal-name`*—Name of the proposal. The proposal name can be up to 32 alphanumeric characters long.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

ipsec-policy

IN THIS SECTION

Syntax

```
ipsec-policy from-zone zone-name to-zone zone-name ipsec-group-vpn vpn-name;
```

Hierarchy Level

```
[edit security]
```

Description

Specifies that matching traffic is checked against rules associated with the specified Group VPN. Exclude and fail-open rules are configured at the `[edit security group-vpn member ipsec vpn` *`vpn-name`*`]` hierarchy level.

Options

from-zone *zone-name*	Specify the incoming zone for Group VPN traffic.
to-zone *zone-name*	Specify the outgoing zone for Group VPN traffic. The `to-zone` zone must include the interface configured with the `group-vpn-external-interface` option at the [`edit security group-vpn member ipsec vpn` *`vpn-name`*] hierarchy level.
ipsec-group-vpn *vpn-name*	Specify the Group VPN to which the traffic applies. Only one Group VPN can be referenced by a specific from-zone/to-zone pair.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

ipsec-sa (Security Group VPN)

IN THIS SECTION

Syntax

```
ipsec-sa name {
    match-policy policy-name {
        destination ip-address/netmask;
        destination-port number;
        protocol number;
        source ip-address/netmask;
        source-port number;
    }
    proposal proposal-name;
}
```

Hierarchy Level

```
[edit security group-vpn server group name]
```

Description

Configure the group SAs to be downloaded to members. There can be multiple group SAs downloaded to group members.

Options

`ipsec-sa` *`name`*—Define the group SAs to be downloaded to members.

- `match-policy` *`policy-name`*—Configure the group policy with source address, source port, destination address, destination port, and protocol.
  - `destination` *`ip-address/netmask`*—Specify the destination IP address to be matched (0.0.0.0/0 for any).
  - `destination-port` *`number`*—Specify the destination port to be matched (0 for any).
  - `protocol` *`number`*—Specify the protocol number to be matched (0 for any).
  - `source` *`ip-address/netmask`*—Specify the source IP address to be matched (0.0.0.0/0 for any).
  - `source-port` *`number`*—Specify the source port to be matched (0 for any)
- `proposal` *`proposal-name`*—Specify the name of the IPsec proposal configured with the `proposal` configuration statement at the `[edit security group-vpn server ipsec]` hierarchy.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | **758**

group (Security Group VPN) | **1514**

ipsec-traffic-selector

IN THIS SECTION

Syntax

```
ipsec-traffic-selector preference pref-value;
```

Hierarchy Level

```
[edit protocol]
```

Description

Configure the global preference value for the ARI-TS (Auto route insertion for traffic selectors) routes added when a IPsec traffic selector based IPsec VPN has been established.

A new category `ipsec-traffic-selector` for Auto route insertion traffic selectors (ARI-TS) route is added in `[edit protocol]` hierarchy for setting global preference value for ARI-TS route.

When a preference value is updated either in global scope or local scope, the already added secure tunnel routes are updated with new preference value.

The preference value update affects the ARI-TS tunnel routes as follows:

- Any changes to the global values impact all traffic selectors configured without a local value.
- Any changes to the local values impact only those specific tunnels.
- Any routes impacted by the change are deleted and added again with the new value.

Default

The default value for `ipsec-traffic-selector` preference value is `5` if not configured.

Options

preference *pref-value* Specify the preferential selection of routes for the same remote IP address or prefix added by different protocols (static, BGP, OSPF, and so on).

- **Range:** 0-4294967295.
- **Default:** 5.

Required Privilege Level

routing—To view this statement in the configuration.

routing-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 22.2R1.

RELATED DOCUMENTATION

traffic-selector | 1656

local-identity

IN THIS SECTION

Syntax

```
local-identity (distinguished-name | hostname identity-hostname | inet identity-ipv4 | inet6
identity-ipv6 | key-id | user-at-hostname identity-user);
```

Hierarchy Level

```
[edit security ike gateway gateway-name]
```

Description

Specify the local IKE identity to send in the exchange with the destination peer to establish communication. If you do not configure a local-identity, the device uses the IPv4 or IPv6 address corresponding to the local endpoint by default.

For Network Address Translation Traversal (NAT-T), both local identity and remote identity must be configured.

Options

- `distinguished-name` *`distinguished name`*—Specify a distinguished name as the identifier for the remote gateway.
- `hostname` *`hostname`*—Specify identity as a fully qualified domain name (FQDN).
- `inet` *`ip-address`*—Specify identity as an IPv4 address.
- `inet6` *`ip-address`*—Specify identity as an IPv6 address.
- `user-at-hostname` *`e-mail-address`*—Specify identity as an e-mail address.
- `key-id` *`sring-key-id`*—Specify key ID in ASCII string.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5. The `inet6` option added in Junos OS Release 11.1.

RELATED DOCUMENTATION

IPsec Overview | 20

manual (Security IPsec)

IN THIS SECTION

Syntax

```
manual {
    authentication {
        algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha1-96);
        key (ascii-text  key | hexadecimal  key );
    }
    encryption {
        algorithm (3des-cbc | aes-128-cbc | aes-128-gcm | aes-192-cbc | aes-256-cbc | aes-256-
gcm | des-cbc);
        key (ascii-text key  | hexadecimal  key );
    }
    external-interface external-interface-name;
    gateway ip-address;
    protocol (ah | esp);
    spi spi-value;
}
```

Hierarchy Level

```
[edit security ipsec vpn vpn-name]
```

Description

Define a manual IPsec security association (SA).

Options

authentication algorithm

Hash algorithm that authenticates packet data. It can be one of the following

- `hmac-md5-96`—Produces a 128-bit digest.
- `hmac-sha-256-128`—Provides data origin authentication and integrity protection. This version of the hmac-sha-256 authenticator produces a 256-bit digest and specifies truncation to 128 bits.
- `hmac-sha1-96`—Hash algorithm that authenticates packet data. It produces a 160-bit digest. Only 96 bits are used for authentication.
- `authentication key`—Type of authentication key. It can be one of the following:
    - `ascii-text` *key*—ASCII text key. For `hmac-md5-96`, the key is 16 ASCII characters; for `hmac-sha1-96`, the key is 20 ASCII characters.
    - `hexadecimal` *key*—Hexadecimal key. For `hmac-md5-96`, the key is 32 hexadecimal characters; for `hmac-sha1-96`, the key is 40 hexadecimal characters.

encryption algorithm

Select the encryption algorithm for the internal Routing-Engine-to-Routing-Engine IPsec security association (SA) configuration. It can be one of the following:

- `des-cbc`—Encryption algorithm with block size of 8 bytes (64 bits) and key size 48 bits.
- `3des-cbc`—Encryption algorithm with block size of 8 bytes (64 bits) and key size of 192 bits.

For `3des-cbc`, we recommend that the first 8 bytes be different from the second 8 bytes, and the second 8 bytes be the same as the third 8 bytes.

- `aes-128-cbc`—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
- `aes-128-gcm`—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
- `aes-192-cbc`—Advanced Encryption Standard (AES) 192-bit encryption algorithm.
- `aes-256-cbc`—Advanced Encryption Standard (AES) 256-bit encryption algorithm.
- `aes-256-gcm`—Advanced Encryption Standard (AES) 256-bit encryption algorithm.
- `encryption key`—Type of encryption key. It can be one of the following:
  - `ascii-text key`—ASCII text key. For the `des-cbc` option, the key contains 8 ASCII characters; for `3des-cbc`, the key contains 24 ASCII characters.
  - `hexadecimal key`—Hexadecimal key. For the `des-cbc` option, the key contains 16 hexadecimal characters; for the `3des-cbc` option, the key contains 48 hexadecimal characters.

external-interface

Specify the outgoing interface for the manual security association

gateway

For a manual security association, specify the IPv4 or IPv6 address of the peer

protocol

Define an IPsec protocol for the manual security association

- Values:
  - ah—Authentication Header protocol
  - esp—ESP protocol (To use the ESP protocol, you must also use the tunnel statement at the [edit security ipsec security-association sa-name mode] hierarchy level)

spi

Configure a security parameter index (SPI) for a security association (SA). An arbitrary value that uniquely identifies which security association (SA) to use at the receiving host (the destination address in the packet).

- **Range:** 256 through 16,639

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 11.1.

Support for `hmac-sha-256-128` added to SRX5400, SRX5600, and SRX5800 devices in Junos OS Release 12.1X46-D20. Support for authentication algorithms (SHA1: hmac-sha1-96 and SHA2: hmac-sha-256-128) in PowerMode IPsec (PMI) mode is introduced for SRX4100, SRX4200, and vSRX Virtual Firewall in Junos OS Release 19.3R1. Support for vSRX Virtual Firewall 3.0 is introduced in Junos OS Release 20.1R1.

Support for cipher algorithms aes-128-cbc, aes-192-cbc, and aes-256-cbc in PowerMode IPsec (PMI) mode is introduced for SRX4100, SRX4200, and vSRX Virtual Firewall in Junos OS Release 19.3R1. Support for vSRX Virtual Firewall 3.0 is introduced in Junos OS Release 20.1R1.

RELATED DOCUMENTATION

IPsec Overview | 20
vpn (Security) | 1662

member (Security Group VPN)

IN THIS SECTION

Syntax

```
member {
    ike {
        gateway gateway-name;
            policy;
            proposal;
            traceoptions;
        }
    ipsec {
        vpn vpn-name {
            df-bit (clear | copy | set);
            exclude rule rule-name {
                source-address ip-address/mask;
                destination-address ip-address/mask;
                application application;
            }
            fail-open rule rule-name {
                source-address ip-address/mask;
                destination-address ip-address/mask;
                application application;
            }
            group id;
            group-vpn-external-interface interface;
            ike-gateway gateway-name;
            recovery-probe;
        }
    }
}
```

Hierarchy Level

```
[edit security group-vpn]
```

Description

Configure group VPN member. A group member encrypts the traffic and is responsible for the actual encryption and decryption of data traffic. A group member is configured with IKE Phase 1 parameters and GC/KS information.

Options

ike*gateway-name* Configure IKE gateway for group VPN member.

policy *policy-name* Configure an IKE policy.

proposal*proposal-name* Define an IKE proposal.

traceoptions Configure group VPN tracing options to aid in troubleshooting the IKE issues.

ipsec Configure IPsec for Phase 2 exchange on the group member.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

mode (Security Group VPN)

IN THIS SECTION

Syntax

```
mode (aggressive | main);
```

Hierarchy Level

```
[edit security group-vpn member ike policy policy-name]
[edit security group-vpn server ike policy policy-name]
```

Description

Define the mode used for Internet Key Exchange (IKE) Phase 1 negotiations. Use aggressive mode only when you need to initiate an IKE key exchange without ID protection, as when a peer unit has a dynamically assigned IP address. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

- IKEv2 protocol does not negotiate using mode configuration.
- The device deletes existing IKE and IPsec SAs when you update the `mode` configuration in the IKE policy.

Options

- `aggressive`—Aggressive mode.
- `main`—Main mode. Main mode is the recommended key-exchange method because it conceals the identities of the parties during the key exchange.

  Configuring `mode main` for group VPN servers or members is not supported when the remote gateway has a dynamic address and the authentication method is `pre-shared-keys`.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5. Support for `group-vpn` hierarchies added in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

policy (Security Group VPN IKE) | 1577

multi-sa

IN THIS SECTION

Syntax

```
multi-sa {
    forwarding-class expedited-forwarding | assured-forwarding | best-effort | network-control;
}
```

Hierarchy Level

```
[edit security ipsec vpn]
```

Description

Negotiate multiple security association (SAs) based on configuration choice. Multiple SAs negotiates with the same traffic selector on the same IKE SA. By negotiating multiple SAs, the peer gateways have more replay windows. If the peer gateways create separate multiple SAs for the configured Forwarding-Classes (FC), then potentially a separate anti-replay window is available for each FC value. With this mapping, even if CoS can reorder packets, reordering is done with in a given multiple SA, thus avoiding packets drop due to the anti-replay checks.

Options

forwarding-class — Forwarding classes (FCs) allow you to group packets for transmission and to assign packets to output queues.

- Values:
  - `expedited-forwarding`—Provides a low-loss, low-latency, low-jitter, assured-bandwidth, end-to-end service.
  - `assured-forwarding`—Provides a group of values you can define and includes four subclasses—AF1, AF2, AF3, and AF4—each with three drop probabilities (low, medium, and high).
  - `best-effort`—Provides no service profile. For the BE forwarding class, loss priority is typically not carried in a class-of-service (CoS) value, and random early detection (RED) drop profiles are more aggressive.
  - `network-control`—This class is typically high priority because it supports protocol control.

Required Privilege Level

security

Release Information

Statement introduced in Junos OS Release 18.2R1.

RELATED DOCUMENTATION

show security ipsec security-associations | 1899

Understanding CoS-Based IPsec VPNs with Multiple IPsec SAs | 553

Example: Configuring CoS-Based IPsec VPNs | 559
vpn (Security) | 1662

ocsp (Security PKI)

IN THIS SECTION

Syntax

```
ocsp {
    connection-failure (disable | fallback-crl);
    disable-responder-revocation-check;
    nonce-payload (enable | disable);
    url ocsp-url;
}
```

Hierarchy Level

```
[edit security pki ca-profile ca-profile-name revocation-check]
```

Description

Configure Online Certificate Status Protocol (OCSP) to check the revocation status of a certificate.

Options

connection-failure (Optional) Specify action to take if there is a connection failure to the OCSP responder. If this option is not configured and there is no response from the OCSP responder, certificate validation will fail.

- **disable** Skip the revocation check if the OCSP responder is not reachable.
- **fallback-crl** Use CRL to check the revocation status of the certificate.

disable-responder-revocation-check (Optional) Disable revocation check for the CA certificate received in an OCSP response. The certificates received in an OCSP response generally have shorter lifetimes and revocation check is not required.

nonce-payload (Optional) Send a nonce payload to prevent replay attack. A nonce payload is sent by default unless it is explicitly disabled. If enabled, the SRX Series Firewall expects OCSP responses to contain a nonce payload, otherwise the revocation check will fail. If OCSP responders are not capable of responding with a nonce payload, disable this option.

- **disable** Explicitly disable the sending of a nonce payload.
- **enable** Enable the sending of a nonce payload. This is the default.

url *ocsp-url* Specify HTTP addresses for OCSP responders. A maximum of two HTTP URL addresses can be configured. If the configured URLs are not reachable, or URLs are not configured, the URL from the certificate being verified is checked.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.1X46-D20.

RELATED DOCUMENTATION

PKI Components In Junos OS | 33

revocation-check (Security PKI) | 1618

pki

IN THIS SECTION

Syntax

```
pki {
    auto-re-enrollment;
    ca-profile ca-profile-name;
    default-trusted-ca-certs (Security);
    traceoptions;
    trusted-ca-group trusted-ca-group-name {
        ca-profiles ca-profiles;
    }
}
```

Hierarchy Level

```
[edit security]
```

Description

Configure an IPsec profile and related options to request digital certificates. The Public Key Infrastructure (PKI) provides an infrastructure for digital certificate management.

Options

auto-re-enrollment Configure the automatic reenrollment of a local end-entity (EE) certificate.

ca-profile ***ca-profile-name*** Configure certificate authority (CA) profile.

default-trusted-ca-certs Configure automatic download of default trusted CA certificates.

traceoptions Configure public key infrastructure (PKI) tracing options.

trusted-ca-group ***trusted-ca-group-name*** Configure trusted certificate authority group.

- **ca-profiles** Name of the CA profiles. You can configure maximum of 20 CA profiles.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5.

`default-trusted-ca-certs` option is added in Junos OS Release 23.2R1.

RELATED DOCUMENTATION

PKI Components In Junos OS | 33

policy (Security Group VPN IKE)

IN THIS SECTION

Syntax

```
policy policy-name {
    description description;
    mode2 (aggressive | main);
    pre-shared-key (ascii-text key | hexadecimal key);
    proposals proposal-name;
}
```

Hierarchy Level

```
[edit security group-vpn member ike]
[edit security group-vpn server ike]
```

Description

Configure an IKE policy. An IKE policy defines a combination of security parameters (IKE proposals) to be used during IKE negotiation. It defines a peer address, the preshared key for the given peer, and the proposals needed for that connection. During the IKE negotiation, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

Options

policy *policy-name* — Name of the IKE policy. The policy name can be up to 32 alphanumeric characters long.

description *description* — Specify descriptive text for an IKE policy.

mode — Define the mode used for Internet Key Exchange (IKE) Phase 1 negotiations.

pre-shared-key — Define a preshared key for an IKE policy. Preshared keys are used to secure the Phase 1 SAs between the root-server and the sub-servers and between the sub-servers and the group members. Ensure that the preshared keys used are strong keys. On the sub-servers, the preshared key configured for the IKEpolicy RootSrv must match the preshared key configured on the root-server, and the preshared key configured for the IKE policy GMs must match the preshared key configured on the group members.

proposals *proposal-name* — Specify up to four Phase 1 proposals for an IKE policy. If you include multiple proposals, use the same Diffie-Hellman group in all of the proposals.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

policy (Security IKE)

IN THIS SECTION

Syntax

```
policy policy-name {
    certificate {
        local-certificate certificate-id;
        peer-certificate-type (pkcs7 | x509-signature);
        policy-oids [ oid ];
        trusted-ca {
            ca-profile ca-profile-name;
            trusted-ca-group trusted-ca-group-name;
        }
    }
    description description;
    mode (aggressive | main);
    pre-shared-key (ascii-text key | hexadecimal key);
    seeded-pre-shared-key (ascii-text key | hexadecimal key);
    proposal-set (basic | compatible | prime-128 | prime-256 | standard | suiteb-gcm-128 | suiteb-gcm-256);
    proposals proposal-name;
    reauth-frequency number;
}
```

Hierarchy Level

```
[edit security ike]
```

Description

IKE policies define a combination of security parameters (IKE proposals) to be used during IKE negotiation, including peer address, the preshared key for the given peer, and the proposals needed for that connection. During the IKE negotiation, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

IKE proposals in the `policy` statement are evaluated in list order, from top to bottom, so when creating the policy, specify the highest priority proposal first, followed by the next highest priority, and so on.

Options

`policy-name`—Name of the IKE policy. The policy name can be up to 32 alphanumeric characters long.

`certificate`—Specify usage of a digital certificate to authenticate the virtual private network (VPN) initiator and recipient.

`description` *`description`*—Specify the description of IKE policy.

`mode`—Define the mode used for Internet Key Exchange (IKE) Phase 1 negotiations. Use aggressive mode only when you need to initiate an IKE key exchange without ID protection, as when a peer unit has a dynamically assigned IP address. IKEv2 protocol does not negotiate using mode configuration. The device deletes existing IKE and IPsec SAs when you update the `mode` configuration in the IKE policy.

- `aggressive`—Aggressive mode.
- `main`—Main mode. Main mode is the recommended key-exchange method because it conceals the identities of the parties during the key exchange.

  Configuring `mode main` for group VPN servers or members is not supported when the remote gateway has a dynamic address and the authentication method is `pre-shared-keys`.

`pre-shared-key`—Define a preshared key for an IKE policy. The device deletes existing IKE and IPsec SAs when you update the `pre-shared-key` configuration in the IKE policy.

- `ascii-text` *`key`*—Specify a string of 1 to 255 ASCII text characters for the key. To include the special characters `( ) [ ] ! & ? |` enclose either the entire key string or the special character in quotation marks; for example `“str)ng”` or `str”)”ng`. Other use of quotation marks within the string is not allowed. With `des-cbc` encryption, the key contains 8 ASCII characters. With `3des-cbc` encryption, the key contains 24 ASCII characters.
- `hexadecimal` *`key`*—Specify a string of 1 to 255 hexadecimal characters for the key. Characters must be hexadecimal digits `0` through `9`, or letters `a` through `f` or `A` through `F`. With `des-cbc` encryption, the key contains 16 hexadecimal characters. With `3des-cbc` encryption, the key contains 48 hexadecimal characters.

`seeded-pre-shared-key`—Define a seeded preshared key in ASCII or hexadecimal format for an IKE policy. The `seeded-pre-shared-key` is a master key that is used to generate the `pre-shared-key` for the peers. Thus each peer will have different `pre-shared-key`. The advantage of this option is that each peer connection to gateway will have different pre-shared key, so if one of the peer's `pre-shared-key` is compromised, then the other peers are not impacted.

The peer preshared keys are generated using the master key configured as `seeded-pre-shared-key` and shared across the peers. To view the peer's pre-shared-key, execute the `show security ike pre-shared-key` command, share and configure the displayed pre-shared key in peer's device as pre-shared-key (in ASCII format). Master key is only configured in the gateway device and not shared to any peer.

You can retrieve the peer preshared key using the `show security ike pre-shared-key user-id` *`peer ike-id`* `master-key` *`master key`* or `show security ike pre-shared-key user-id` *`peer ike-id`* `gateway` *`gateway name`* command.

- `ascii-text` *`key`*—Configure a string of 1 to 255 ASCII text characters for the key. To include the special characters `( ) [ ] ! & ? |` enclose either the entire key string or the special character in quotation marks; for example `“str)ng”` or `str”)”ng`. Other use of quotation marks within the string is not allowed.
- `hexadecimal` *`key`*—Specify a string of 1 to 255 hexadecimal characters for the key. Characters must be hexadecimal digits `0` through `9`, or letters `a` through `f` or `A` through `F`.

`proposal-set`—Specify a set of default Internet Key Exchange (IKE) proposals.

`proposals` *`proposal-name`*—Specify up to four Phase 1 proposals for an IKE policy. If you include multiple proposals, use the same Diffie-Hellman group in all of the proposals.

`reauth-frequency` *`number`*—Configure the reauthentication frequency to trigger a new IKEv2 reauthentication. Reauthentication creates a new IKE SA, creates new child SAs within the IKE SA, and then deletes the old IKE SA. This option is disabled by default. umber of IKE rekeys that occurs before reauthentication occurs. If `reauth-frequency` is `1`, reauthentication occurs every time there is an IKE rekey. If `reauth-frequency` is `2`, reauthentication occurs at every other IKE rekey. If `reauth-frequency` is `3`, reauthentication occurs at every third IKE rekey.

- **Default:** 0 (disable)
- **Range:** 0-100

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5.

Support for `suiteb-gcm-128` and `suiteb-gcm-256` options added in Junos OS Release 12.1X45-D10.

Support for `policy-oids` option added in Junos OS Release 12.3X48-D10.

Support for `trusted-ca` option added in Junos OS Release 18.1R1.

Support for `reauth-frequency` option added in Junos OS Release 15.1X49-D60.

Support for `seeded-pre-shared-key` option added in Junos OS Release 21.1R1.

RELATED DOCUMENTATION

IPsec Overview | 20

Example: Configuring AutoVPN with Pre-Shared Key | 1294

policy (Security IPsec)

IN THIS SECTION

Syntax

```
policy policy-name {
    description description;
    perfect-forward-secrecy keys (group1 | group14 | group19 | group2 | group20 | group24 |
group5 | group15 | group16 | group21);
    proposal-set (basic | compatible | prime-128 | prime-256 | standard | suiteb-gcm-128 |
suiteb-gcm-256);
    proposals proposal-name;
}
```

Hierarchy Level

```
[edit security ipsec]
```

Description

Define an IPsec policy. An IPsec policy defines a combination of security parameters (IPsec proposals) used during IPsec negotiation. It defines Perfect Forward Secrecy (PFS) and the proposals needed for the connection.

Options

name Name of the IPsec policy.

description Enter descriptive text for an IPsec policy.

perfect-forward-secrecy keys Specify Perfect Forward Secrecy (PFS) as the method that the device uses to generate the encryption key. PFS generates each new encryption key independently from the previous key. The device deletes existing IPsec SAs when you update the `perfect-forward-secrecy` configuration in the IPsec policy.

- Values:
  - `group1`—768-bit Modular Exponential (MODP) algorithm.
  - `group2`—1024-bit MODP algorithm.
  - `group5`—1536-bit MODP algorithm.
  - `group14`—2048-bit MODP group.
  - `group15`—3072-bit MODP algorithm.
  - `group16`—4096-bit MODP algorithm.
  - `group19`—256-bit random Elliptic Curve Groups modulo a Prime (ECP groups) algorithm.

- `group20`—384-bit random ECP groups algorithm.
- `group21`—521-bit random ECP groups algorithm.
- `group24`—2048-bit MODP Group with 256-bit prime order subgroup.

proposal-set Define a set of default IPsec proposals.

- Values:
  - basic—IPsec basic proposal set. esp-des-sha and esp-des-md5.
    - Encapsulating Security Payload (ESP) protocol
    - Encryption algorithm—DES-CBC encryption algorithm
    - Authentication algorithm—SHA1 or MD5 authentication algorithm
  - compatible—IPsec compatible proposal set. esp-3des-sha, esp-3des-md5, esp-des-sha, and esp-des-md5.
    - ESP protocol
    - Encryption algorithm—3DES-CBC or DES-CBC encryption algorithm
    - Authentication algorithm—SHA1 or MD5 authentication algorithm
  - `prime-128`—Provides the following proposal set:
    - Encapsulating Security Payload (ESP) protocol
    - Encryption algorithm—Advanced Encryption Standard Galois/Counter mode (AES-GCM)128-bit
    - Authentication algorithm—None (AES-GCM provides both encryption and authentication)

    This option is not supported on Group VPNv2.
  - `prime-256`—Provides the following proposal set:
    - ESP protocol
    - Encryption algorithm—AES-GCM 256-bit

  - Authentication algorithm—None (AES-GCM provides both encryption and authentication)

  This option is not supported on Group VPNv2.

- `standard`—esp-3des-sha and esp-aes128-sha
  - ESP protocol
  - Encryption algorithm—3DES-CBC or AES-CBC 128-bit encryption algorithm
  - Authentication algorithm—SHA1 authentication algorithm
- `suiteb-gcm-128`—Provides the following proposal set:
  - ESP protocol
  - Encryption algorithm—AES-GCM 128-bit
  - Authentication algorithm—None (AES-GCM provides both encryption and authentication)

  This option is not supported on Group VPNv2.
- `suiteb-gcm-256`—Provides the following proposal set:
  - ESP protocol
  - Encryption algorithm—AES-GCM 256-bit
  - Authentication algorithm—None (AES-GCM provides both encryption and authentication)

  This option is not supported on Group VPNv2.

proposals ***proposal-name***

Specify up to four Phase 2 proposals for an IPsec policy. If you include multiple proposals, use the same Diffie-Hellman group in all of the proposals.

Proposals are evaluated in the order they appear on the list, from top down, so specify the highest priority first, followed by the next highest priority, and so on.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5.

Support for group 14 is added in Junos OS Release 11.1.

Support for `group14` options added in Junos OS Release 11.1.

Support for `group19`, `group20`, and `group24` options added in Junos OS Release 12.1X45-D10.

`group15`, `group16`, and `group21` options introduced in Junos OS Release 19.1R1 on SR5000 line with `junos-ike` package installed.

Support for `suiteb-gcm-128` and `suiteb-gcm-256` options added in Junos OS Release 12.1X45-D10. Support for `prime-128` and `prime-256` options added in Junos OS Release 15.1X49-D40.

Starting in Junos OS Release 20.2R1, we've changed the help text description as `NOT RECOMMENDED` for the CLI options `group1`, `group2`, and `group5` for devices running IKED with `junos-ike` package installed.

Support for `group15`, `group16`, and `group21` options added in Junos OS Release 20.3R1 on vSRX Virtual Firewall instances with `junos-ike` package installed.

Support for `group15`, `group16`, and `group21` options added in Junos OS Release 21.1R1 on vSRX Virtual Firewall 3.0 instances with `junos-ike` package installed.

RELATED DOCUMENTATION

IPsec Overview | 20

Installing Junos IKE package

power-mode-ipsec

IN THIS SECTION

Syntax

```
power-mode-ipsec;
```

Hierarchy Level

```
[edit security flow (Security Flow)]
```

Description

Enable PowerMode IPsec. processing. PMI is a new mode of operation that provides IPsec performance improvements.

For SRX4100, SRX4200 devices running Junos OS Release 18.4R1, SRX4600 devices running Junos OS Release 20.4R1, and vSRX Virtual Firewall instances running Junos OS Release 18.3R1, you can enable or disable the PMI. Starting in Junos OS Release 21.1R1, you can enable or disable the PMI on MX-SPC3 services card.

If you use Junos OS Release 18.3R1, you must reboot the device for the configuration to take effect.

From Junos OS Release 18.4R1 and later, you don’t need reboot the device after enabling or disabling this feature.

Packets cannot go through the PMI when firewall or advanced security services are combined with IPsec. Hence, PMI must not be used when firewall or advanced security services are combined with IPsec.

Required Privilege Level

flow-tap

Release Information

Statement introduced in Junos OS Release 18.3R1.

RELATED DOCUMENTATION

Improving IPsec Performance with PowerMode IPsec | 1383

profile (Juniper Secure Connect)

IN THIS SECTION

Syntax

```
profile realm-name {
    access-profile access-profile;
    client-config client-config;
    compliance {
        pre-logon compliance-rule;
    }
    description description;
    ipsec-vpn ipsec-vpn;
}
```

Hierarchy Level

```
[edit security remote-access]
```

Description

Configure remote user connection profiles for the Juniper Secure Connect clients.

The remote access profiles allow you to deploy connection settings for the remote users by pushing the configuration file on the client devices. You can create multiple profiles and set one of the profiles as the default profile.

NOTE: Starting in Junos OS Release 23.1R1, we've hidden the `default-profile` option at the `[edit security remote-access]` hierarchy level. In releases before Junos OS Release 23.1R1, you use this option to specify one of the remote-access profiles as the default profile in Juniper Secure Connect. But with changes to the format of remote-access profile names, we no longer require the `default-profile` option.

We've deprecated `default-profile` option—rather than immediately removing it—to provide backward compatibility and a chance to make your existing configuration conform to the changed configuration. You'll receive a warning message if you continue to use the `default-profile` option in your configuration. However existing deployments are not affected if you modify the current configuration. See default-profile (Juniper Secure Connect).

Each remote access profile includes a *realm-name* mapping to an URL either in *FQDN/RealmName* or *FQDN* format, authentication settings, VPN settings, and client configurations. You can create different remote access profiles for different names or functions.

Example—You can create a configuration profile for the engineering department, and another for the human resource department. You name the profile for engineering department and human resource department as *ra.example.com/engineering* and *ra.example.com/hr* respectively.

When a Juniper Secure connect remote user selects a connection profile such as *ra.example.com/engineering*, the SRX Series Firewall receives the configuration request and selects a remote-access profile with same name —that is—*ra.example.com/engineering* for pushing the configuration on client device.

Options

realm-name Set realm-name as remote-access profile name. This is the profile identifier in FQDN/RealmName format.

Examples:

ra.example.com/hr, if FQDN is *ra.example.com* and Realm name is *hr*.

ra.example.com/engineering, if FQDN is *ra.example.com* and Realm name is *engineering*.

ra.example.com, if FQDN is *ra.example.com* and Realm name is empty.

Specify an IP address if you do not have an *FQDN* (*192.168.1.10/hr* or *192.168.1.10)*.

access-profile Select the access profile for authentication and accounting for clients.

client-config Select the client configuration object.

compliance Select pre-logon compliance rule object name.

description Text description of the remote access profile.

ipsec-vpn Select the IPsec VPN policy object used for IKE and IPsec proposals.

Required Privilege Level

security

Release Information

Statement introduced in Junos OS Release 20.3R1.

Compliance option introduced in Junos OS Release 23.1R1.

Realms format changed to FQDN/RealmName or FQDN in Junos OS Release 23.1R1.

RELATED DOCUMENTATION

Juniper Secure Connect Administrator Guide

proposal (Security Group VPN Member IKE)

IN THIS SECTION

Syntax

```
proposal proposal-name {
    authentication-algorithm (sha-256 | sha-384);
    authentication-method pre-shared-keys;
    description description;
    dh-group (group14 | group24);
    encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
    lifetime-seconds seconds;
}
```

Hierarchy Level

```
[edit security group-vpn member ike]
```

Description

Define an IKE proposal. You can configure one or more IKE proposals. Each proposal is a list of IKE attributes to protect the IKE connection between the IKE host and its peer.

Options

`proposal` *proposal-name*—Name of the IKE proposal. The proposal name can be up to 32 alphanumeric characters long.

`authentication-algorithm`—Configure the Internet Key Exchange (IKE) authentication algorithm. Hash algorithm that authenticates packet data. It can be one of the following algorithms:

- `sha-256`—Produces a 256-bit digest. This is the default value.
- `sha-384`—Produces a 384-bit digest.

`authentication-method` *pre-shared-keys*—Specify the method the device uses to authenticate the source of Internet Key Exchange (IKE) messages. The `pre-shared-keys` option refers to a preshared key, which is a secret key shared between the two peers, is used during authentication to identify the peers to each other. The same key must be configured for each peer. This is the default method.

`description` *description*—Specify descriptive text for an IKE proposal.

`dh-group`—Specify the IKE Diffie-Hellman group for key establishment.

- `group14`—2048-bit group. This is the default value.
- `group24`—2048-bit, 256 bit subgroup. Support for the `group24` option added in Junos OS Release 15.1X49-D30 for vSRX Virtual Firewall.

`encryption-algorithm`—Configure an encryption algorithm for an IKE proposal.

- `aes-128-cbc`—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
- `aes-192-cbc`—AES 192-bit encryption algorithm.

- `aes-256-cbc`—AES 256-bit encryption algorithm.

`lifetime-seconds` *seconds*—Specify the lifetime (in seconds) of an IKE or IPsec security association (SA) for group VPN. When the SA expires, it is replaced by a new SA and security parameter index (SPI) or terminated.

- **Range:** 180 through 86,400 seconds
- **Default:** 3600 seconds

The device does not delete existing IPsec SAs when you update the `authentication-algorithm`, `authentication-method`, `dh-group`, and `encryption-algorithm` configuration in the IKE proposal.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

proposal (Security Group VPN Server IKE)

IN THIS SECTION

Syntax

```
proposal proposal-name {
    authentication-algorithm (sha-256 | sha-384);
    authentication-method pre-shared-keys;
    description description;
    dh-group (group14 | group24);
    encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
}
```

Hierarchy Level

```
[edit security group-vpn server ike]
```

Description

Define an IKE proposal for group VPN server. You can configure one or more IKE proposals. Each proposal is a list of IKE attributes to protect the IKE connection between the IKE host and its peer.

Options

`proposal` *`proposal-name`*—Name of the IKE proposal. The proposal name can be up to 32 alphanumeric characters long.

`authentication-algorithm`—Configure the Internet Key Exchange (IKE) authentication algorithm. Hash algorithm that authenticates packet data. It can be one of the following algorithms:

- `sha-256`—Produces a 256-bit digest. This is the default value.
- `sha-384`—Produces a 384-bit digest.

`authentication-method` *pre-shared-keys*—Specify the method the device uses to authenticate the source of Internet Key Exchange (IKE) messages. The `pre-shared-keys` option refers to a preshared key, which is a secret key shared between the two peers, is used during authentication to identify the peers to each other. The same key must be configured for each peer. This is the default method.

`description` *description*—Specify descriptive text for an IKE proposal.

`dh-group`—Specify the IKE Diffie-Hellman group for key establishment.

- `group14`—2048-bit group. This is the default value.
- `group24`—2048-bit, 256 bit subgroup. Support for the `group24` option added in Junos OS Release 15.1X49-D30 for vSRX Virtual Firewall.

`encryption-algorithm`—Configure an encryption algorithm for an IKE proposal.

- `aes-128-cbc`—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
- `aes-192-cbc`—AES 192-bit encryption algorithm.
- `aes-256-cbc`—AES 256-bit encryption algorithm.

The device does not delete existing IPsec SAs when you update the `authentication-algorithm`, `authentication-method`, `dh-group`, and `encryption-algorithm` configuration in the IKE proposal.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

ike (Security Group VPN Server) | 1531

proposal (Security Group VPN Server IPsec)

IN THIS SECTION

Syntax

```
proposal proposal-name {
    authentication-algorithm (hmac-sha-256-128);
    description description;
    encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
    lifetime-seconds seconds;
}
```

Hierarchy Level

```
[edit security group-vpn server ipsec]
```

Description

Define an IPsec proposal. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

Options

proposal-name—Name of the IPsec proposal.

`authentication-algorithm` *hmac-sha-256-128*—Configure the IPsec authentication algorithm. Produces a 256-bit digest, truncated to 128 bits. This is the default value.

`description` *description*—Text the description of IPsec proposal.

`encryption-algorithm`—Configure an encryption algorithm. The device deletes existing IPsec SAs when you update the `encryption-algorithm` configuration in the IPsec proposal.

- `aes-128-cbc`—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
- `aes-192-cbc`—AES 192-bit encryption algorithm.
- `aes-256-cbc` —AES 256-bit encryption algorithm. This is the default value.

`lifetime-seconds` *seconds*—Specify the lifetime (in seconds) of an IPsec security association (SA) for group VPN. When the SA expires, it is replaced by a new SA and security parameter index (SPI) or terminated. Specify a value from 180 to 86,400 seconds. The default is 3600 seconds.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

ipsec (Security Group VPN Server) | 1553

proposal (Security IKE)

IN THIS SECTION

Syntax

```
proposal proposal-name {
    authentication-algorithm (md5 | sha-256 | sha-384| sha1 | sha-512);
    authentication-method(certificates | dsa-signatures | ecdsa-signatures-256 | ecdsa-
signatures-384 | pre-shared-keys | rsa-signatures | ecdsa-signatures-521);
    description description;
    dh-group (group1 | group14 | group19 | group2 | group20 | group24 | group5 | group15 |
group16 | group21);
    encryption-algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);
    lifetime-seconds seconds;
}
```

Hierarchy Level

```
[edit security ike]
```

Description

Define an IKE proposal.

Options

proposal-name—Name of the IKE proposal. The proposal name can be up to 32 alphanumeric characters long.

`authentication-algorithm`—Configure the Internet Key Exchange (IKE) authentication hash algorithm that authenticates packet data. It can be one of the following algorithms:

- `md5`—Produces a 128-bit digest.
- `sha-256`—Produces a 256-bit digest.
- `sha-384`—Produces a 384-bit digest.
- In Power Mode IPSec mode and in normal mode—
  - `sha1`—Produces a 160-bit digest.
  - `sha-512`—Produces a 512-bit digest.

The device deletes existing IPsec SAs when you update the `authentication-algorithm` configuration either in the IKE proposal or IPsec proposal.

`authentication-method`—Specify the method the device uses to authenticate the source of Internet Key Exchange (IKE) messages. The `pre-shared-keys` option refers to a preshared key, which is a key for encryption and decryption that both participants must have before beginning tunnel negotiations. The other options refer to types of digital signatures, which are certificates that confirm the identity of the certificate holder. The device deletes existing IPsec SAs when you update the `authentication-method` configuration in the IKE proposal.

- `certificates`—You can establish the IKEv2 and IPsec SA tunnels irrespective of the type of certificate used on initiator and responder. The `authentication-method certificates` option cannot be used with IKEv1.

- `dsa-signatures`—Specify that the Digital Signature Algorithm (DSA) is used.

- `ecdsa-signatures-256`—Specify that the Elliptic Curve DSA (ECDSA) using the 256-bit elliptic curve secp256r1, as specified in the *Federal Information Processing Standard (FIPS) Digital Signature Standard (DSS) 186-3*, is used.

- `ecdsa-signatures-384`—Specify that the ECDSA using the 384-bit elliptic curve secp384r1, as specified in the *FIPS DSS 186-3*, is used.

- `pre-shared-keys`—Specify that a preshared key, which is a secret key shared between the two peers, is used during authentication to identify the peers to each other. The same key must be configured for each peer. This is the default method.

- `rsa-signatures`—Specify that a public key algorithm, which supports encryption and digital signatures, is used.

- `ecdsa-signatures-521`—Specify that the ECDSA using the 521-bit elliptic curve secp521r1 is used.

`description` *`description`*—Text the description of IKE proposal.

`dh-group`—Specify the IKE Diffie-Hellman group.

`encryption-algorithm`—Configure an encryption algorithm for an IKE proposal.

`lifetime-seconds` *`seconds`*—Specify the lifetime (in seconds) of an IKE security association (SA). When the SA expires, it is replaced by a new SA and security parameter index (SPI) or terminated.

- **Range:** 180 through 86,400 seconds

- **Default:** 28,800 seconds

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5.

Support for `dh-group group 14` and `dsa-signatures` added in Junos OS Release 11.1.

Support for `sha-384`, `ecdsa-signatures-256`, `ecdsa-signatures-384`, `group19`, `group20`, and `group24` options added in Junos OS Release 12.1X45-D10.

Support for `ecdsa-signatures-256` and `ecdsa-signatures-384` options added in Junos OS Release 12.1X45-D10.

Support for `sha-512`, `group15`, `group16`, `group21`, and `ecdsa-signatures-521` options added in Junos OS Release 19.1R1 on SRX5000 line with `junos-ike` package installed.

Support for authentication algorithm (SH1: hmac-sha1-96) added to vSRX Virtual Firewall in Junos OS Release 19.3R1 for Power Mode IPSec mode, along with the existing support in normal mode.

Support for `group15`, `group16`, and `group21` options added in Junos OS Release 20.3R1 on vSRX Virtual Firewall instances with `junos-ike` package installed.

Support for `group15`, `group16`, and `group21` options added in Junos OS Release 21.1R1 on vSRX Virtual Firewall 3.0 instances with `junos-ike` package installed.

Support for `certificates` option added in Junos OS Release 22.4R1 on MX240, MX480, and MX960 in USF mode, SRX1500, SRX4200, SRX4600, SRX5400, SRX5600, SRX5800, and vSRX Virtual Firewall 3.0 running iked process.

RELATED DOCUMENTATION

IPsec Overview | 20

ike

Configuring an IKE Proposal for Dynamic SAs

proposal (Security IPsec)

IN THIS SECTION

Syntax

```
proposal proposal-name {
    authentication-algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha-256-96 | hmac-sha-384 |
hmac-sha-512 | hmac-sha1-96);
    description description;
    encryption-algorithm (3des-cbc | aes-128-cbc | aes-128-gcm | aes-192-cbc | aes-192-gcm |
aes-256-cbc | aes-256-gcm | des-cbc);
    extended-sequence-number;
    lifetime-kilobytes kilobytes;
    lifetime-seconds seconds;
    protocol (ah | esp);
}
```

Hierarchy Level

```
[edit security ipsec]
```

Description

Define an IPsec proposal. An IPsec proposal lists protocols and algorithms (security services) to be negotiated with the remote IPsec peer.

Options

proposal-name

Name of the IPsec proposal.

authentication-algorithm

Configure the IPsec authentication algorithm. Authentication algorithm is the hash algorithm that authenticates packet data. It can be one of six algorithms:

- Values:

  The hash algorithm to authenticate data can be one of the following:

  - `hmac-md5-96`—Produces a 128-bit digest.
  - `hmac-sha-256-128`—Provides data origin authentication and integrity protection. This version of the hmac-sha-256 authenticator produces a 256-bit digest and specifies truncation to 128 bits.
  - `hmac-sha1-96`—Hash algorithm that authenticates packet data. It produces a 160-bit digest. Only 96 bits are used for authentication.
  - `hmac-sha-512`—Produces a 512-bit digest.
  - `hmac-sha-384`—Produces a 384-bit digest.
  - `hmac-sha-256-96`—HMAC-SHA-256-96 authentication algorithm (non-RFC compliant)

description

Text description of IPsec proposal

encryption-algorithm

Define encryption algorithm. The device deletes existing IPsec SAs when you update the `encryption-algorithm` configuration in the IPsec proposal.

- Values:

  - `3des-cbc`—Encryption algorithm with block size of 8 bytes (64 bits) and key size of 192 bits.
  - `aes-128-cbc`—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
  - `aes-128-gcm`—AES Galois/Counter Mode (GCM) 128-bit encryption algorithm.

    For an IKE proposal, AES 128-bit authenticated encryption algorithm is supported with IKEv2 only. When this option is used, `aes-128-gcm` should be configured at the [`edit security ipsec proposal` *`proposal-name`*] hierarchy level, and the `authentication-algorithm` option should not be configured at the [`edit security ike proposal` *`proposal-name`*] hierarchy level.

When `aes-128-gcm`, `aes-192-gcm`, or `aes-256-gcm` encryption algorithms are configured in the IPsec proposal, it is not mandatory to configure AES-GCM encryption algorithm in the corresponding IKE proposal.

- `aes-192-cbc`—AES 192-bit encryption algorithm.
- `aes-192-gcm`—AES GCM 192-bit encryption algorithm.
- `aes-256-cbc`—AES 256-bit encryption algorithm.
- `aes-256-gcm`—AES GCM 256-bit encryption algorithm.

  For an IKE proposal, AES 256-bit authenticated encryption algorithm is supported with IKEv2 only. When this option is used, `aes-256-gcm` should be configured at the [`edit security ipsec proposal` *`proposal-name`*] hierarchy level, and the `authentication-algorithm` option should not be configured at the [`edit security ike proposal` *`proposal-name`*] hierarchy level.
- `des-cbc`—Encryption algorithm with block size of 8 bytes (64 bits) and key size 48 bits.

extended-sequence-number

Use the `extended-sequence-number` option to enable ESN support. ESN allows IPsec to use 64-bit sequence numbers for the sequence number. If ESN is not enabled, 32-bit sequence number will be used by default. Ensure ESN is not enabled when anti-replay is disabled.

lifetime-kilobytes

Specify the lifetime (in kilobytes) of an IPsec security association (SA). If this statement is not configured, the number of kilobytes used for the SA lifetime is unlimited.

- **Range:** 64 through 1,048,576 kilobytes

lifetime-seconds

Lifetime in seconds.

- **Range:** 180 through 86400
- **Default:** 3600 seconds

protocol

Define the IPsec protocol for a manual or dynamic security association (SA).

- Values:

- ah—Authentication header
- esp—Encapsulated Security Payload header

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced before Junos OS Release 7.4.

`extended-sequence-number` option introduced in Junos OS Release 19.4R1.

Starting in Junos OS Release 20.2R1, we've changed the help text description as `NOT RECOMMENDED` for the CLI options `hmac-md5-96`, `hmac-sha1-96`, `3des-cbc`, and `des-cbc` for devices running IKED with `junos-ike` package installed.

`hmac-sha-512` and `hmac-sha-384` options introduced in Junos OS Release 19.1R1 on SRX5000 line with SRX5K-SPC3 card.

Support for `aes-128-gcm`, `aes-192-gcm`, and `aes-256-gcm` options added in Junos OS Release 15.1X49-D70 for vSRX Virtual Firewall.

Support for `aes-128-gcm`, `aes-192-gcm`, and `aes-256-gcm` options added in Junos OS Release 12.1X45-D10.

Support for `hmac-sha-256-128` added to SRX5400, SRX5600, and SRX5800 devices in Junos OS Release 12.1X46-D20.

RELATED DOCUMENTATION

Configuring an IPsec Proposal for an ES PIC

Installing Junos IKE package

proposal-set (Security IKE)

IN THIS SECTION

Syntax

```
proposal-set (basic | compatible | prime-128 | prime-256 | standard | suiteb-gcm-128 | suiteb-
gcm-256);
```

Hierarchy Level

```
[edit security ike policy policy-name]
```

Description

Specify a set of default Internet Key Exchange (IKE) proposals.

The `prime-128` and `prime-256` proposal sets require IKEv2 and certificate-based authentication.

Options

- `basic`—Includes a basic set of two IKE proposals:
  - Proposal 1—Preshared key, Data Encryption Standard (DES) encryption, and Diffie-Hellman (DH) group 1 and Secure Hash Algorithm 1 (SHA-1) authentication.
  - Proposal 2—Preshared key, DES encryption, and DH group 1 and Message Digest 5 (MD5) authentication.
- `compatible`—Includes a set of four commonly used IKE proposals:
  - Proposal 1—Preshared key, triple DES (3DES) encryption, and Diffie-Hellman (DH) group 2 (DH group 2) and SHA-1 authentication.
  - Proposal 2—Preshared key, 3DES encryption, and DH group 2 and MD5 authentication.
  - Proposal 3—Preshared key, DES encryption, and DH group 2 and SHA-1 authentication.
  - Proposal 4—Preshared key, DES encryption, and DH group 2 and MD5 authentication.
- `prime-128`—Provides the following proposal set (this option is not supported on Group VPNv2):
  - Authentication method—Elliptic Curve Digital Signature Algorithm (ECDSA) 256-bit signatures.
  - Diffie-Hellman Group—19.
  - Encryption algorithm—Advanced Encryption Standard (AES) 128-bit Galois/Counter Mode (GCM).
  - Authentication algorithm—None (AES-GCM provides both encryption and authentication).

  When this option is used, `prime-128` should also be configured at the [`edit security ipsec policy` *`policy-name`* `proposal-set`] hierarchy level.
- `prime-256`—Provides the following proposal set (this option is not supported on Group VPNv2):
  - Authentication method—ECDSA 384-bit signatures.
  - Diffie-Hellman Group—20.
  - Encryption algorithm—AES 256-bit GCM.
  - Authentication algorithm—None (AES-GCM provides both encryption and authentication).

  When this option is used, `prime-256` should also be configured at the [`edit security ipsec policy` *`policy-name`* `proposal-set`] hierarchy level.
- `standard`—Includes a standard set of two IKE proposals:

  - Proposal 1— Preshared key, 3DES encryption, and DH group 2 and SHA-1 authentication.
  - Proposal 2—Preshared key, AES 128-bit encryption, and DH group 2 and SHA-1 authentication.

- `suiteb-gcm-128`—Provides the following Suite B proposal set (this option is not supported on Group VPNv2):
  - Authentication method—ECDSA 256-bit signatures
  - Diffie-Hellman Group—19
  - Encryption algorithm—Advanced Encryption Standard (AES) 128-bit Galois Counter Mode (GCM)

    GCM mode is used instead of CBC.
  - Authentication algorithm—SHA-256
- `suiteb-gcm-256`—Provides the following Suite B proposal set (this option is not supported on Group VPNv2):
  - Authentication method—ECDSA 384-bit signatures
  - Diffie-Hellman Group—20
  - Encryption algorithm—AES 256-bit GCM

    GCM mode is used instead of CBC.
  - Authentication algorithm—SHA-384

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5. Support for `suiteb-gcm-128` and `suiteb-gcm-256` options added in Junos OS Release 12.1X45-D10. Support for `prime-128` and `prime-256` options added in Junos OS Release 15.1X49-D40.

Starting in Junos OS Release 20.2R1, we've changed the help text description as `NOT RECOMMENDED` for the CLI options `basic`, `compatible`, and `standard` for devices running IKED with `junos-ike` package installed.

RELATED DOCUMENTATION

IPsec Overview | 20

Installing Junos IKE package

remote-access (Juniper Secure Connect)

IN THIS SECTION

Syntax

```
remote-access {
    client-config name {
        application-bypass {
            term name {
                description description;
                protocol protocol;
                domain-name domain-name
     }
}
        biometric-authentication;
        certificate {
            no-expiry-warning;
            no-pin-request-per-connection;
            warn-before-expiry days;
        }
        connection-mode (always | manual);
        dead-peer-detection {
            interval seconds;
            threshold threshold;
        }
        no-dead-peer-detection;
        no-eap-tls;
        no-tcp-encap;
        windows-logon {
            auto-dialog-open;
            disconnect-at-logoff;
            domain domain;
            eap-auth;
            flush-credential-at-logoff;
            lead-time-duration seconds;
            mode (automatic | manual);
        }
    }
    compliance
        pre-logon name {
        term term-name {
             match {
                 platform {
                    (android | ios | macos | windows) {
                        (app-version | os-version) {
                            (equal | greater-than | greater-than-or-equal | less-than | less-
than-or-equal) version;
                    }
                }
            }
                hostname value;
                ms-domain value;
                ms-workgroup value;
                deviceid value;
            }
             action (accept | reject);
    }
}
    default-profile default-profile;
    global-options {
        auth-token-valid-time seconds;
    }
    profile realm-name {
        access-profile access-profile;
        client-config client-config;
        compliance {
            pre-logon compliance-rule;
        description description;
        ipsec-vpn ipsec-vpn;
    }
    traceoptions {
        file <filename> <files files> <match match> <size size> <(world-readable | no-world-
readable)>;
        flag name;
        level (brief | detail | extensive | verbose);
        no-remote-trace;
    }
}
```

Hierarchy Level

```
[edit security]
```

Description

Configure remote access settings.

You must configure the remote client settings on SRX Series Firewall to facilitate auto configuration for Juniper Secure Connect remote clients.

When a remote client downloads Juniper Secure Connect application, the application establishes an HTTPS connection with the security device. All authenticated clients fetch the configuration file from the security device and establish a VPN tunnel. This step eliminates the need for the remote clients to configure parameters for certificate identifier parameters, remote access client settings, and IKE and IPsec parameters on their device to establish a VPN connection.

Options

client-config Define Juniper Secure Connect remote client configuration parameters.

compliance Configure the compliance rules for the Juniper Secure Connect client's connection request

default-profile Configure default profile. On your security device, you must specify one of the remote-access profiles as the default profile.

> **NOTE**: Starting in Junos OS Release 23.1R1, we've hidden the `default-profile` option at the `[edit security remote-access]` hierarchy level. In releases before Junos OS Release 23.1R1, you use this option to specify one of the remote-access profiles as the default profile in Juniper Secure Connect. But with changes to the format of remote-access profile names, we no longer require the `default-profile` option.
>
> We've deprecated `default-profile` option—rather than immediately removing it—to provide backward compatibility and a chance to make your existing configuration conform to the changed configuration. You'll receive a warning message if you continue to use the `default-profile` option in your configuration. However existing deployments are not affected if you modify the current configuration. See default-profile (Juniper Secure Connect).

global-options Define global parameters for Juniper Secure Connect remote access configuration.

profile Configure remote user connection profiles for the Juniper Secure Connect clients.

traceoptions Configure remote access tracing operations for Juniper Secure Connect.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security

Release Information

Statement introduced in Junos OS Release 20.3R1.

Support for prelogon compliance rule is added in Junos OS Release 23.1R1.

Support for application bypass is added in Junos OS Release 23.1R1.

Usage of `default-profile` options is not allowed starting Junos OS Release 23.1R1.

RELATED DOCUMENTATION

Juniper Secure Connect Administrator Guide

remote-identity

IN THIS SECTION

Syntax

```
remote-identity {
    distinguished-name {
        container container-string;
        wildcard wildcard-string;
    }
    hostname hostname;
    inet ip-address;
    inet6 ipv6-address;
    key-id;
    user-at-hostname e-mail-address;
}
```

Hierarchy Level

```
[edit security ike gateway gateway-name]
```

Description

Specify the remote IKE identity to exchange with the destination peer to establish communication. If you do not configure a remote-identity, the device uses the IPv4 or IPv6 address corresponding to the remote endpoint by default.

For Network Address Translation Traversal (NAT-T), both remote identity and local identity must be configured.

Options

- `distinguished-name`—Specify identity as the distinguished name (DN) from the certificate. If there is more than one certificate on the device, use the `security ike gateway` *gateway-name* `policy` *policy-name* `certificate local-certificate` *certificate-id*.

  Optional container and wildcard strings can be specified:

  - `container` *container-string*—Specify a string for the container.
  - `wildcard` *wildcard-string*—Specify a string for the wildcard.

- `hostname` *hostname*—Specify identity as a fully qualified domain name (FQDN).
- `inet` *ip-address*—Specify identity as an IPv4 address.
- `inet6` *ipv6-address*—Specify identity as an IPv6 address.

- `key-id` *`string-key-id`*—Specify the key ID in ASCII sring.
- `user-at-hostname` *`e-mail-address`*—Specify identity as an e-mail address.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 11.4.

RELATED DOCUMENTATION

IPsec Overview | 20

replay-attacks

IN THIS SECTION

Syntax

```
replay-attacks {
    threshold value;
}
```

Hierarchy Level

```
[edit security alarms potential-violation]
```

Description

Raise a security alarm when the device detects a replay attack. A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed.

Default

Replay attacks do not raise security alarms.

Options

- `threshold` *`value`*—Number of reply attacks up to which an alarm is not raised. When the configured number is exceeded, an alarm is raised.
- **Range:** 1 through 100,00,00,000.
- **Default:** 1000

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 11.2.

RELATED DOCUMENTATION

IPsec Overview | 20

Monitoring VPN Traffic | 1351

revocation-check (Security PKI)

IN THIS SECTION

Syntax

```
revocation-check {
    crl:
```

```
    disable;
    ocsp:
    use-crl;
    use-ocsp;
}
```

Hierarchy Level

```
[edit security pki ca-profile ca-profile-name]
```

Description

Specify the method the device uses to verify the revocation status of digital certificates.

Options

crl Only certificate revocation list (CRL) is supported. A CRL is a time-stamped list identifying revoked certificates, which is signed by a CA and made available to the participating IPsec peers on a regular periodic basis.

You should also specify the location (URL) to retrieve the CRL (HTTP or LDAP). By default, the URL is empty and uses CDP information embedded in the CA certificate.

For Example: `set security pki ca-profile ms-ca revocation-check crl url http://labsrv1.labdomain.com/CertEnroll/LABDOMAIN.crl`

The URL can include the server-name or port information such as, ldap://<ip-or-fqdn>:<port>). If the port number is missing, HTTP uses port 80, or LDAP uses port 443. Currently, you can configure only one URL. We do not support for configuring backup URL.

By default, `crl` is enabled. Local certificates are being validated against certificate revocation list (CRL) even when CRL check is disabled. This can be stopped by disabling the CRL check through the Public Key Infrastructure (PKI) configuration. When CRL check is disabled, PKI will not validate local certificate against CRL.

disable Disable verification of status of digital certificates.

ocsp Configure Online Certificate Status Protocol (OCSP) to check the revocation status of a certificate.

use-crl Specify the CRL as the method to check the revocation status of a certificate. CRL is the default method.

When you enable this option, you choose CRL as a method to verify the revocation status of digital certificates.

use-ocsp Specify the Online Certificate Status Protocol (OCSP) as the method to check the revocation status of a certificate. CRL is the default method.

When you enable this option, you choose OCSP as a method to verify the revocation status of digital certificates.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5. Support for `ocsp`, `use-crl`, and `use-ocsp` options added in Junos OS Release 12.1X46-D20.

RELATED DOCUMENTATION

PKI Components In Junos OS | 33

ca-profile (Security PKI) | 1455

security-association

IN THIS SECTION

Syntax

```
security-association sa-name {
    manual {
        direction bidirectional {
            authentication {
                algorithm (hmac-md5-96 | hmac-sha1-96 | hmac-sha-256 | hmac-sha-384 | hmac-
sha-512);
                key {
                    ascii-text key;
                    hexadecimal key;
                }
            }
            encryption {
                algorithm (3des-cbc | des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc);
                key {
                    ascii-text key;
                    hexadecimal key;
                }
            }
            protocol (ah | esp);
            spi spi-value;
        }
    }
    mode transport;
}
```

Hierarchy Level

```
[edit security ipsec]
```

Description

Configure a manual IPsec security association (SA) to be applied to an OSPF or OSPFv3 interface or virtual link. IPsec can provide authentication and confidentiality to OSPF or OSPFv3 routing packets.

Options

sa-name Name of the SA.

description Specify a text description for the SA.

direction Direction of the manual SA. For this feature, the direction must be `bidirectional`. Decrypt and authenticate the incoming and outgoing traffic using the same algorithm, keys, or SPI in both directions, unlike inbound and outbound SAs that use different attributes in both directions.

- **Values:** `algorithm`—Hash algorithm that authenticates packet data. It can be one of the following:
  - `hmac-md5-96`—Produces a 128-bit digest. This is the default.
  - `hmac-sha1-96`—Produces a 160-bit digest.
  - `hmac-sha-256`—Produces a 256-bit digest.
  - `hmac-sha-384`—Produces a 384-bit digest.

- `hmac-sha-512`—Produces a 512-bit digest.

  Starting in Junos OS Release 22.2R1, MX240, MX480, and MX960 with MX-SPC3, SRX Series Firewalls and vSRX Virtual Firewall running iked process supports all the listed authentication algorithms.

  `key`—Type of authentication key. It can be one of the following:

  - `ascii-text` *`key`*—ASCII text key. For `hmac-md5-96`, the key is 16 ASCII characters; for `hmac-sha1-96`, the key is 20 ASCII characters.

  - `hexadecimal` *`key`*—Hexadecimal key. For `hmac-md5-96`, the key is 32 hexadecimal characters; for `hmac-sha1-96`, the key is 40 hexadecimal characters.

- **Values:** `encryption`—Configure an encryption algorithm and key for a manual Security Association (SA). It can be one of the following:

  - `algorithm`—Select the encryption algorithm for the internal Routing-Engine-to-Routing-Engine IPsec security association (SA) configuration.

    - `des-cbc`—Encryption algorithm with block size of 8 bytes (64 bits) and key size 48 bits.

    - `3des-cbc`—Encryption algorithm with block size of 8 bytes (64 bits) and key size of 192 bits.

      For `3des-cbc`, we recommend that the first 8 bytes be different from the second 8 bytes, and the second 8 bytes be the same as the third 8 bytes.

    - `aes-128-cbc`—Advanced encryption algorithm that has a key size of 16 bytes; its key size is 128 bits long.

    - `aes-192-cbc`—Advanced encryption algorithm that has a key size of 24 bytes; its key size is 192 bits long.

    - `aes-256-cbc`—Advanced encryption algorithm that has a key size of 32 bytes; its key size is 256 bits long.

    Starting in Junos OS Release 22.2R1, MX240, MX480, and MX960 with MX-SPC3, SRX Series Firewalls and vSRX Virtual Firewall running iked process supports all the listed encryption algorithms.

  - `key`—Type of encryption key. It can be one of the following:

  - `ascii-text key`—ASCII text key. For the `des-cbc` option, the key contains 8 ASCII characters; for `3des-cbc`, the key contains 24 ASCII characters.

  - `hexadecimal key`—Hexadecimal key. For the `des-cbc` option, the key contains 16 hexadecimal characters; for the `3des-cbc` option, the key contains 48 hexadecimal characters.

protocol Define the IPsec protocol for a manual security association (SA). The protocol can be one of the following:

- `ah`—Authentication Header protocol. If you configure AH protocol, it is mandatory to configure the authentication algorithm and the key.

- `esp`—Encapsulating Security Payload (ESP) protocol. This is the default.

  If you configure ESP protocol, it is mandatory to configure either authentication algorithm or encryption algorithm or both. If you did not configure ESP protocol and did not configure either authentication or encryption algorithm, then we do not provide authentication or encryption support.

spi *spi-value* Configure the security parameter index (SPI) for a security association (SA). An arbitrary value that uniquely identifies which SA to use at the receiving host (the destination address in the packet).

- **Range:** 256 through 16,639

mode SA mode. For this feature, the mode must be `transport`.

Required Privilege Level

view-level—To view this statement in the configuration.

control-level—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.1X46-D20.

Authentication algorithm configuration options, hmac-md5-96, hmac-sha1-96, hmac-sha-256, hmac-sha-384, and hmac-sha-512 are added in Junos OS Release 22.2R1 for MX240, MX480, and MX960 with MX-SPC3, SRX Series Firewalls and vSRX Virtual Firewall running iked process.

Encryption algorithm configuration options, des-cbc, 3des-cbc, aes-128-cbc, aes-192-cbc, and aes-256-cbc are added in Junos OS Release 22.2R1 for MX240, MX480, and MX960 with MX-SPC3, SRX Series Firewalls and vSRX Virtual Firewall running iked process.

RELATED DOCUMENTATION

Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls | 199

server (Security Group VPN)

IN THIS SECTION

Syntax

```
server {
    group name {
        anti-replay-time-window milliseconds;
        description description;
        group-id number;
        ike-gateway [gateway-name];
        ipsec-sa name {
            match-policy policy-name {
                destination ip-address/netmask;
                destination-port number;
                protocol number;
                source ip-address/netmask;
                source-port number;
            }
            proposal proposal-name;
        }
        member-threshold number;
        server-cluster {
            ike-gateway gateway-name;
            retransmission-period seconds;
            server-role (root-server | sub-server);
        }
        server-member-communication {
            certificate certificate-id;
            communication-type unicast;
            encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
            lifetime-seconds seconds;
            number-of-retransmission number;
            retransmission-period seconds;
            sig-hash-algorithm (sha-256 | sha-384);
        }
    }
    ike {
        gateway  gateway-name {
            address ip-address ;
            dead-peer-detection {
                always-send;
                interval seconds;
                threshold number;
            }
            dynamic {
                (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
            }
            ike-policy policy-name;
            local-address ip-address;
            local-identity {
                (hostname hostname | inet ip-address | user-at-hostname e-mail-address);
            }
            remote-identity {
                (hostname [hostname] | inet ip-address | user-at-hostname e-mail-address);
            }
            routing-instance routing-instance;
        }
        policy policy-name {
            description text;
            mode (aggressive | main);
            pre-shared-key (ascii-text key | hexadecimal key);
            proposals [proposal-name];
        }
        proposal proposal-name {
            authentication-algorithm (sha-256 | sha-384);
            authentication-method pre-shared-keys;
            description description;
            dh-group (group14 | group24);
            encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
        }
    }
    ipsec {
        proposal proposal-name {
            authentication-algorithm hmac-sha-256-128;
            description description;
            encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
            lifetime-seconds seconds;
        }
    }
    traceoptions {
        file {
            filename;
            files number;
            match regular-expression;
            size maximum-file-size;
            (world-readable | no-world-readable);
        }
        flag flag;
        gateway-filter {
            local-address ip-address;
            remote-address ip-address;
        }
        level (all | error | info | notice | verbose | warning);
        no-remote-trace;
    }
}
```

Hierarchy Level

```
[edit security group-vpn]
```

Description

Configure group VPN server. You can configure the following on the group server:

- Phase 1 IKE SA for group members
- Phase 2 IPsec proposal
- Group identifier, group members, server-member communications, and group policies to be downloaded to members
- Group VPN trace options

Options

gateway *gateway-name* — Configure IKE gateway for group VPN server.

ike — Configure Phase 1 security association (SA) with a member on the group server.

ipsec — Configure an IPsec proposal for Phase 2 exchange on the group server.

traceoptions — Configure group VPN tracing options to aid in troubleshooting the IKE issues.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

server-cluster (Security Group VPN Server)

IN THIS SECTION

Syntax

```
server-cluster {
    ike-gateway gateway-name;
    retransmission-period seconds;
    server-role (root-server | sub-server);
}
```

Hierarchy Level

```
[edit security group-vpn server group name]
```

Description

Configure the Group Domain of Interpretation (GDOI) group controller/key server (GCKS) cluster for the specified group. All servers in a group VPN server cluster must be SRX Series Firewalls.

Options

ike-gateway *gateway-name*

(Required) Specify the name of the IKE gateway for the local device in the group server cluster. IKE gateways are configured at the `[edit security group-vpn server ike]` hierarchy level.

If the local device is a root-server, the IKE gateway name must be a sub-server in the cluster; up to four sub-server IKE gateways can be specified.

If the local device is a sub-server, the IKE gateway name must be the root-server.

retransmission-period *seconds*

(Optional) Specify the time after which the root-server retransmits a `cluster-update` message if it has not received an acknowledgement from a sub-server.

- **Range:** 2 to 60 seconds.
- **Default:** 10 seconds.

server-role

(Required) Assign the role of the local device in the group server cluster, either `root-server` or `sub-server`. Only one device in the cluster can be configured as the root-server. You can configure up to four other devices as a sub-server in a group server cluster.

You must ensure that there is only one root-server at any time for a group VPN server cluster.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

Understanding Group VPNv2 Server Clusters | 818

group (Security Group VPN) | 1514

server-member-communication (Security Group VPN Server)

IN THIS SECTION

Syntax

```
server-member-communication {
    certificate certificate-id;
    communication-type (unicast);
    encryption-algorithm (aes-128-cbc | aes-192-cbc | aes-256-cbc);
    lifetime-seconds seconds;
    number-of-retransmission number;
    retransmission-period seconds;
    sig-hash-algorithm (sha-256 | sha-384);
}
```

Hierarchy Level

```
[edit security group-vpn server group name]
```

Description

Enable and configure server to member communication. When these options are configured, group members receive new keys before current keys expire. Starting with Junos OS Release 15.1X49-D80, the minimum value that you can configure for the `lifetime-seconds` option is 300 seconds instead of 180 seconds.

Options

- `certificate` *`certificate-id`*—Specify the certificate identification. Only RSA keys are supported.
- `communication-type`—Configure `unicast` (the default).
- `encryption-algorithm`—Encryption used for communications between the group server and group member. Specify `aes-128-cbc`, `aes-192-cbc`, or `aes-256-cbc`.
- `lifetime-seconds` *`seconds`*—Lifetime, in seconds, of the key encryption key (KEK). Specify a value from 300 to 86,400. The default is 3600 seconds.

- `number-of-retransmission` *number*—For unicast communications, the number of times the group server retransmits messages to a group member when there is no reply. Specify a value from 0 to 60. The default is 2.

- `retransmission-period` *seconds*—The time period between a transmission and the first retransmission when there is no reply from the group member. Specify a value from 2 to 60. The default is 10 seconds.

- `sig-hash-algorithm`—Authentication algorithm used to authenticate the group member to the group server. Specify `sha-256` or `sha-384`.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

group (Security Group VPN) | 1514

session-affinity

IN THIS SECTION

Syntax

```
session-affinity ipsec
```

Hierarchy Level

```
[edit security flow load-distribution]
```

Description

Enable VPN session affinity. In session affinity feature, we've optimized tunnel redistribution. After tunnel redistribution, the data path might not be optimal, hence we recommend that you enable VPN session affinity to ensure that the data path is optimized. During optimization, the current data path experiences a higher packet delay until it is fully optimized.

This feature is supported on SRX5400, SRX5600, and SRX5800 devices. By default, VPN session affinity is disabled.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 11.4R5.

Starting with Junos OS Release 15.1X49-D10, IPsec session affinity is supported for IPsec tunnel-based traffic by the SRX5K-MPC3-100G10G (IOC3) and the SRX5K-MPC3-40G10G (IOC3) for SRX5400, SRX5600, and SRX5800 devices through improved flow module and session cache.

RELATED DOCUMENTATION

IPsec Overview | 20

tcp-encap

IN THIS SECTION

Syntax

```
tcp-encap {
    profile profile-name;
        ssl-profile ssl-profile-name;
        log ;
    }
    traceoptions {
        file filename {
            files number;
            match regular-expression;
            size maximum-file-size;
            (world-readable | no-world-readable);
        }
        flag (all | configuration | session | tunnel);
        level (all | error | info | notice | verbose | warning);
        no-remote-trace’
    }
```

Hierarchy Level

```
[edit security]
```

Description

Specify TCP encapsulation operations for a remote access client to a remote access gateway on an SRX Series Firewall to support IPsec messages encapsulated within a TCP connection.

Options

profile *profile-name* — Configure a TCP encapsulation profile for a remote access client to a remote access gateway on an SRX Series Firewall to define the data encapsulation operation.

- ssl-profile *ssl-profile-name* — Specify the SSL termination profile that is configured at the `[edit services ssl termination profile]` hierarchy level. This parameter is required for NCP Exclusive Remote Access Client of Full SSL Session.

- log — Enable logging for remote access client connections.

traceoptions — Configure TCP encapsulation tracing options.

Required Privilege Level

view-level—To view this statement in the configuration.

control-level—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 15.1X49-D80.

Support for the `ssl-profile` option added in Junos OS Release 18.1R1.

RELATED DOCUMENTATION

Understanding SSL Remote Access VPNs with NCP Exclusive Remote Access Client

traceoptions (Juniper Secure Connect)

IN THIS SECTION

Syntax

```
traceoptions {
    file <filename> <files files> <match match> <size size> <(world-readable | no-world-
readable)>;
    flag name;
    level (brief | detail | extensive | verbose);
    no-remote-trace;
}
```

Hierarchy Level

```
[edit security remote-access]
```

Description

Configure remote access tracing operations for Juniper Secure Connect. By default, messages are written to **/var/log/*file-name*** file. The default file name if not configured is ravpn_trace.

Options

filename

Name of file in which to write trace information.

files

Maximum number of trace files.

- **Default:** 3
- **Range:** 2 through 1000

match

Regular expression for lines to be logged.

no-world-readable

Don't allow any user to read the log file.

size

Maximum trace file size.

- **Default:** 128k
- **Range:** 10 KB through the maximum file size supported on your system.

world-readable

Allow any user to read the log file.

flag *flag*—Tracing operation to perform

Tracing operation to perform.

- Values:
  - all—Trace everything
  - cli-configuration—Trace CLI configuration events

level *level*

Tracing level. The following values are supported:

- brief—Brief debugging output
- detail—Detailed debugging output
- extensive—Extensive debugging output
- verbose—Verbose debugging output

no-remote-trace

Disable remote tracing.

Required Privilege Level

trace

Release Information

Statement introduced in Junos OS Release 20.3R1.

RELATED DOCUMENTATION

Juniper Secure Connect Administrator Guide

traceoptions (Security Group VPN)

IN THIS SECTION

Syntax

```
traceoptions {
    file {
        filename;
        files number;
        match regular-expression;
        size maximum-file-size;
        (world-readable | no-world-readable);
    }
    flag flag (all | certificates | config | database | general | high-availability | ike | next-
hop-tunnels | parse | policy-manager | routing-socket | thread | timer);
    gateway-filter {
        local-address ip-address;
        remote-address ip-address;
    }
    level (all | error | info | notice | verbose | warning);
    no-remote-trace;
}
```

Hierarchy Level

```
[edit security group-vpn member ike]
[edit security group-vpn server]
```

Description

Configure group VPN tracing options to aid in troubleshooting the IKE or server issues. This helps troubleshoot one or multiple tunnels negotiation by standard tracefile configuration. Tracing allows the user to view the detailed packet exchange and the negotiation information. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances.

Options

- `file`—Configure the trace file options.
  - *`filename`*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory `/var/log`.
  - `files` *`number`*—Maximum number of trace files. When a trace file named *`trace-file`* reaches its maximum size, it is renamed to *`trace-file`*`.0`, then *`trace-file`*`.1`, and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.

    If you specify a maximum number of files, you also must specify a maximum file size with the `size` option and a filename.

    Range: 2 through 1000 files

    Default: 10 files
  - `match` *`regular-expression`*—Refine the output to include lines that contain the regular expression.
  - `size` *`maximum-file-size`*—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *`trace-file`* reaches this size, it is renamed *`trace-file`*`.0`. When the `trace-file` again reaches its maximum size, *`trace-file`*`.0` is renamed *`trace-file`*`.1` and *`trace-file`* is renamed *`trace-file`*`.0`. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the `files` option and filename.

Syntax: *x* k to specify KB, *x*m to specify MB, or *x*g to specify GB

Range: 10 KB through 1 GB

Default: 128 KB

  - `world-readable | no-world-readable`—By default, log files can be accessed only by the user who configures the tracing operation. The `world-readable` option enables any user to read the file. To explicitly set the default behavior, use the `no-world-readable` option.

- `flag`—Trace operation to perform. To specify more than one trace operation, include multiple `flag` statements.

  - `all`—Trace all activity.
  - `certificates`—Trace certificate-related activity.
  - `config`—Trace configuration activity.
  - `database`—Trace SA-related database activity.
  - `general`—Trace general activity.
  - `high-availability`—Trace high-availability operations.
  - `ike`—Trace IKE protocol activity.
  - `next-hop-tunnels`—Trace next-hop tunnel operations.
  - `parse`—Trace configuration processing.
  - `policy-manager`—Trace IKE callback activity.
  - `routing-socket`—Trace routing socket activity.
  - `thread`—Trace thread processing.
  - `timer`—Trace timer activity.

- `gateway-filter`—Configure debugging for the tunnel between the group VPN server and a group member. This option is configured on a group VPN server or member.

  - `local-address`—When configured on a server, the IP address of the group VPN server. When configured on a member, the IP address of the group VPN member.
  - `remote-address`—When configured on a server, the IP address of the group VPN member. When configured on a member, the IP address of the group VPN server.

- level—Set the level of debugging.
  - all—Match all levels.
  - error—Match error conditions.
  - info—Match informational messages.
  - notice—Match conditions that should be handled specifically.
  - verbose—Match verbose messages.
  - warning—Match warning messages.
- no-remote-trace—Disable remote tracing.

Required Privilege Level

trace—To view this statement in the configuration.

trace-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.2. Support for gateway-filter option for the [edit security group-vpn member ike] hierarchy level added in Junos OS Release 15.1X49-D30 for vSRX Virtual Firewall.

RELATED DOCUMENTATION

Group VPNv2 Overview | **758**

traceoptions (Security IKE)

IN THIS SECTION

Syntax

```
traceoptions {
    file {
        filename;
        files number;
        match regular-expression;
        size maximum-file-size;
        (world-readable | no-world-readable);
    }

    level (critical | error | terse | warning | detail);
    flag flag (all | certificates | config | database | general | high-availability | ike | next-
hop-tunnels | parse | policy-manager | routing-socket | thread | timer);
    no-remote-trace;
    rate-limit messages-per-second;
}
```

Hierarchy Level

```
[edit security ike]
```

Description

Configure IKE tracing options to aid in troubleshooting the IKE issues. This helps troubleshoot one or multiple tunnels negotiation by standard tracefile configuration. IKE tracing allows the user to view the detailed packet exchange and the negotiation information in Phase 1 and Phase 2. IKE tracing is not enabled by default. By default , all IKE or IPsec negotiations are logged into /var/log/kmd. But user can also specify customized file name while configuring the IKE traceoptions.

Options

- file—Configure the trace file options.
    - `filename`—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory `/var/log`.

      Default: kmd

    - `files` *`number`*—Maximum number of trace files. When a trace file named *`trace-file`* reaches its maximum size, it is renamed to *`trace-file`*`.0`, then *`trace-file`*`.1`, and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.

      If you specify a maximum number of files, you also must specify a maximum file size with the `size` option and a filename.

      Range: 2 through 1000 files

      Default: 10 files

    - `match` *`regular-expression`*—Refine the output to include lines that contain the regular expression.
    - `size` *`maximum-file-size`*—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *`trace-file`* reaches this size, it is renamed *`trace-file`*`.0`. When the `trace-file` again reaches its maximum size, *`trace-file`*`.0` is renamed *`trace-file`*`.1` and *`trace-file`* is renamed *`trace-file`*`.0`. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the `files` option and filename.

Syntax: *x* `k` to specify KB, *x* `m` to specify MB, or *x* `g` to specify GB

Range: 10 KB through 1 GB

Default: 1024 KB

- `world-readable | no-world-readable`—By default, log files can be accessed only by the user who configures the tracing operation. The `world-readable` option enables any user to read the file. To explicitly set the default behavior, use the `no-world-readable` option.

- `level`—Specify the log levels.
  - `critical`—Log single point failures which needs your immediate attention
  - `error`—Log fatal application errors
  - `terse`—Log syslog messages
  - `warning`—Log recoverable errors
  - `detail`—Log all operational information
- `flag`—Trace operation to perform. To specify more than one trace operation, include multiple `flag` statements.
  - `all`—Trace all iked process modules activity
  - `certificates`—Trace certificate-related activity
  - `config`—Trace configuration download processing
  - `database`—Trace VPN-related database activity
  - `general`—Trace general activity
  - `high-availability`—Trace high-availability operations
  - `ike`—Trace IKE protocol activity
  - `next-hop-tunnels`—Trace next-hop tunnels operations
  - `parse`—Trace VPN parsing activity
  - `policy-manager`—Trace iked callback activity
  - `routing-socket`—Trace routing socket activity

  - `thread`—Trace thread processing
  - `timer`—Trace timer activity

  By default, the `flag` statement is not set. You need to explicitly configure the `flag` statement to perform trace operation.

- `no-remote-trace`—Set remote tracing as disabled.
- `rate-limit` *`messages-per-second`*—Configure the incoming rate of trace messages.

  Range: 0 through 4,294,967,295

  Default: 0

Required Privilege Level

trace—To view this statement in the configuration.

trace-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5.

`level` options introduced in Junos OS Release 21.1R1.

RELATED DOCUMENTATION

IPsec Overview | 20

ike (Security) | 1524

traceoptions (Security IPsec)

IN THIS SECTION

Syntax

```
traceoptions {
    flag flag;
}
```

Hierarchy Level

```
[edit security ipsec]
```

Description

Configure IPsec tracing options. Trace operations track IPsec events and record them in a log file in the /var/log directory.

Trace operations are written to the trace file **/var/log/kmd**.

Options

- flag—To specify more than one trace operation, include multiple flag statements.
  - all—Trace with all flags enabled
  - next-hop-tunnel-binding—Trace next-hop tunnel binding events
  - packet-drops—Trace packet drop activity
  - packet-processing—Trace data packet processing events
  - security-associations—Trace security association (SA) management events

Required Privilege Level

trace—To view this statement in the configuration.

trace-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5.

RELATED DOCUMENTATION

IPsec Overview | 20

traceoptions (Security PKI)

IN THIS SECTION

Syntax

```
traceoptions {
    file {
        filename;
        files number;
        match regular-expression;
        size maximum-file-size;
        (world-readable | no-world-readable);
    }
    flag {
        all;
        certificate-verification;
        online-crl-check;
    }
    no-remote-trace;
}
```

Hierarchy Level

```
[edit security pki]
```

Description

Configure public key infrastructure (PKI) tracing options. To specify more than one trace option, include multiple flag statements. Trace option output is recorded in the /var/log/pkid file.

Options

- `file`—Configure the trace file options.
  - *`filename`*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory `/var/log`. By default, the name of the file is the name of the process being traced.
  - `files` *`number`*—Maximum number of trace files. When a trace file named *`trace-file`* reaches its maximum size, it is renamed to *`trace-file`*`.0`, then *`trace-file`*`.1`, and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.

    If you specify a maximum number of files, you also must specify a maximum file size with the `size` option and a filename.

    Range: 2 through 1000 files

    Default: 10 files
  - `match` *`regular-expression`*—Refine the output to include lines that contain the regular expression.
  - `size` *`maximum-file-size`*—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *`trace-file`* reaches this size, it is renamed *`trace-file`*`.0`. When the `trace-file` again reaches its maximum size, *`trace-file`*`.0` is renamed *`trace-file`*`.1` and *`trace-file`* is renamed *`trace-file`*`.0`. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

    If you specify a maximum file size, you also must specify a maximum number of trace files with the `files` option and a filename.

    Syntax: *`x`* `K` to specify KB, *`x`* `m` to specify MB, or *`x`* `g` to specify GB

    Range: 10 KB through 1 GB

    Default: 128 KB
  - `world-readable` | `no-world-readable`—By default, log files can be accessed only by the user who configures the tracing operation. The `world-readable` option enables any user to read the file. To explicitly set the default behavior, use the `no-world-readable` option.

- flag—Trace operation to perform. To specify more than one trace operation, include multiple flag statements.
  - all—Trace with all flags enabled
  - certificate-verification—Trace PKI certificate verification events
  - online-crl-check—Trace PKI online certificate revocation list (CRL) events
- no-remote-trace—Set remote tracing as disabled.

Required Privilege Level

trace—To view this statement in the configuration.

trace-control—To add this statement to the configuration.

Release Information

Statement modified in Junos OS Release 8.5.

RELATED DOCUMENTATION

PKI Components In Junos OS | 33

traceoptions (TCP Encapsulation)

IN THIS SECTION

Syntax

```
traceoptions {
    file filename {
        files number;
        match regular-expression;
        size maximum-file-size;
        (world-readable | no-world-readable);
    }
    flag (all | configuration | session | tunnel);
    level (all | error | info | notice | verbose | warning);
    no-remote-trace;
}
```

Hierarchy Level

```
[edit security tcp-encap]
```

Description

Configure TCP encapsulation tracing options.

Options

file Configure the trace file options.

- *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory `/var/log`.
- `files` *number*—Maximum number of trace files. When a trace file named *trace-file* reaches its maximum size, it is renamed to *trace-file*`.0`, then *trace-file*`.1`, and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.

  If you specify a maximum number of files, you also must specify a maximum file size with the `size` option and a filename.

  Range: 2 through 1000 files

  Default: 10 files
- `match` *regular-expression*—Refine the output to include lines that contain the regular expression.
- `size` *maximum-file-size*—Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches its maximum size, it is renamed *trace-file*`.0`. When *trace-file*`.0` reaches its maximum size, it is renamed *trace-file*`.1` and *trace-file* is renamed *trace-file*`.0`. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

  If you specify a maximum file size, you also must specify a maximum number of trace files with the `files` option and filename.

  Syntax: *x* `k` to specify KB, *x* `m` to specify MB, or *x* `g` to specify GB

  Range: 10 KB through 1 GB

  Default: 128 KB
- `world-readable | no-world-readable`—By default, log files can be accessed only by the user who configures the tracing operation. The `world-readable` option enables any user to read the file. To explicitly set the default behavior, use the `no-world-readable` option.

flag

Trace operation to perform. To specify more than one trace operation, include multiple flag statements.

- `all`—Trace all activity.
- `configuration`—Trace configuration events.
- `session`—Trace session related events.
- `tunnel`—Trace tunnel events.

level

Set the level of debugging.

- `all`—Match all levels.
- `error`—Match error conditions.
- `info`—Match informational messages.
- `notice`—Match conditions that should be handled specifically.
- `verbose`—Match verbose messages.
- `warning`—Match warning messages.

no-remote-trace Disable remote tracing.

Required Privilege Level

view-level—To view this statement in the configuration.

control-level—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 15.1X49-D80.

RELATED DOCUMENTATION

> Understanding SSL Remote Access VPNs with NCP Exclusive Remote Access Client
>
> tcp-encap | **1635**

traffic-selector

IN THIS SECTION

Syntax

```
traffic-selector traffic-selector-name {
    local-ip ip-address/netmask;
    remote-ip ip-address/netmask;
    preference pref_value;
    protocol protocol_name/protocol_id;
    source-port low-high;
    destination-port low-high;
    metric metric_value;
    description description_value;
    term term_name {
    local-ip ip-address/netmask;
    remote-ip ip-address/netmask;
    protocol protocol_name/protocol_id;
    source-port low-high;
    destination-port low-high;
                      }
}
```

Hierarchy Level

```
[edit security ipsec vpn vpn-name]
```

Description

A traffic selector is an agreement between IKE peers to permit traffic through a tunnel, if the traffic matches a specified pair of local IP address range, remote IP address range, source port range, destination port range, and protocol. This functionality is supported only for IKEv2.

In the Junos OS Releases earlier to 21.1R1, we support one pair of local IP prefix and remote IP prefix per IPsec tunnel for traffic filtering through IPsec tunnel. From Junos OS Release 21.1R1 onwards, you can configure multiple sets of local IP prefix, remote IP prefix, source port range, destination port range, and protocol for traffic selection.

This means, multiple sets of IP address ranges, port ranges, and protocols can be part of same traffic selector as defined in RFC 7296. In this functionality, concept of term is introduced within the traffic-selectors. Each term defines a set of local IP range, remote IP range, source port range, destination port range, and protocol. All the terms combined will be part of single IPsec SA. The terms in a single traffic selector can have both IPv4 and IPv6 address. Hence a single IPsec SA has both IPv4 and IPv6 as both local and remote IP addresses. A maximum of 200 terms are supported in each traffic selector.

When you configure multiple traffic selectors, each traffic selector leads to a separate negotiation that results in the multiple IPsec tunnels. But, if you configure multiple terms under one traffic selector, this configuration results in single IPsec SA negotiation with multiple IP prefixes, ports, and protocols.

It is mandatory to configure atleast one local IP prefix and one remote IP prefix for a traffic selector. Other parameters are optional.

If multiple traffic selectors have overlapping routes, a tie breaker of routing metric is used for the forwarding decision.

To install the required Junos package for supporting this functionality on your SRX Series Firewall, use the command `request system software add optional://junos-ike.tgz`.

For backward compatibility, we support configuring IP prefixes directly under the `[edit security ipsec vpn` *`vpn-name`* `traffic-selector` *`traffic-selector-name`*`]` hierarchy.

Use `[edit security ipsec vpn` *`vpn-name`* `traffic-selector` *`traffic-selector-name`* `term` *`term-name`*`]` hierarchy level to configure multiple sets of IP address ranges, port ranges, and protocols for the same traffic selector as defined in RFC 7296.

You should not configure same values for different traffic selectors for the same IKE gateway. This is not a valid traffic selector configuration. If you configure multiple traffic selectors with the same values, then depending on the peer configuration there might be unintended high CPU utilization.

Options

local-ip *ip-address/netmask*

A local IP address or a local subnetwork protected by the local VPN device.

remote-ip *ip-address/netmask*

A remote IP address or a remote subnetwork protected by the peer VPN device.

preference *pref_value*

Local preference value of the traffic selector for a particular `ipsec vpn` *`vpn-name`* that overrides the value specified at global scope.

- **Range:** 0-4294967295.
- **Default:** 5.

term *term_name*

Define a set of local IP range, remote IP range, source port range, destination port range, and protocol. All the terms combined will be part of single IPsec SA. A maximum of 200 terms are supported in each traffic selector. It is optional to configure this parameter.

protocol *protocol_name/protocol_id*

Transport protocol list for a traffic selector for an IPsec tunnel. It is optional to configure this parameter. In case protocol is not configured, then 'any' protocol is assumed to be configured.

- **Range:** Protocol id can range from 0 to 255.

source-port *low-high*

Source port range from lower to higher range port numbers. It is optional to configure this parameter. If no port is configured but only protocol is configured, port 'any' will be assumed for source port ranges for that protocol.

- **Range:** 1 to 65535

destination-port *low-high*

Destination port range from lower to higher range port numbers. It is optional to configure this parameter. If no port is configured but only protocol is configured, port 'any' will be assumed for destination port ranges for that protocol.

- **Range:** 1 to 65535

metric *metric_value*

Tie breaker when multiple traffic selectors have overlapping routes, to decide the most preferred path. It is optional to configure this parameter.

description *description_value*

Traffic selector description. It is optional to configure this parameter. It is optional to configure this parameter.

- **Range:** 0 to 80 characters

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 12.1X46-D10.

`term`, `protocol`, `source-port`, `destination-port`, `metric`, and `description` options introduced in Junos OS Release 21.1R1.

`preference` *`pref_value`* option introduced in Junos OS Release 22.2R1.

RELATED DOCUMENTATION

IPsec Overview | **20**

vpn (Security) | **1662**

ipsec-traffic-selector | **1559**

verify-path

IN THIS SECTION

Syntax

```
verify-path {
    destination-ip ip-address;
    packet-size bytes;
}
```

Hierarchy Level

```
[edit security ipsec vpn vpn-name vpn-monitor]
```

Description

Verify the IPsec datapath before the secure tunnel (st0) interface is activated and route(s) associated with the interface are installed in the Junos OS forwarding table. This configuration is useful in network topologies where there is a transit firewall located between the VPN tunnel endpoints, and where IPsec data traffic that uses active routes for an established VPN tunnel on the st0 interface might be blocked by the transit firewall.

When this option is configured, the source interface and destination IP addresses that can be configured for VPN monitor operation are not used for IPsec datapath verification. The source for the ICMP requests in the IPsec datapath verification is the local tunnel endpoint.

When IPsec datapath verification is configured, the following actions occur:

1. Upon the establishment of the VPN tunnel, an ICMP request is sent to the peer tunnel endpoint to verify the IPsec datapath.

   The peer tunnel endpoint must be reachable by VPN monitor ICMP requests and must be able to respond to the ICMP request. While the datapath verification is in progress, “`V`” is displayed in the VPN Monitoring field in the `show security ipsec security-association detail` command output.

2. The `st0` interface is activated only when a response is received from the peer.

   The `show interface st0.`*`x`* command output shows the st0 interface status during and after the datapath verification: `Link-Layer-Down` before the verification finishes and `Up` after the verification finishes successfully.

3. If no ICMP response is received from the peer, another ICMP request is sent at the configured VPN monitor interval (the default is 10 seconds) until the VPN monitor threshold (the default is 10 times) is reached.

   If the verification does not succeed, the KMD_VPN_DOWN_ALARM_USER system log entry indicates the reason as a VPN monitoring verify-path error. The error is logged under tunnel events in the `show security ipsec security-association detail` command output. The `show security ipsec tunnel-events-statistics` command displays the number of times the error occurred.

   VPN monitor interval and threshold values are configured with `vpn-monitor-options` at the [`edit security ipsec`] hierarchy level.

4. If no ICMP response is received from the peer after the VPN monitor threshold is reached, the established VPN tunnel is brought down and the VPN tunnel is renegotiated.

Options

destination-ip *ip-address* Original, untranslated IP address of the peer tunnel endpoint that is behind a NAT device. This IP address must not be the NAT translated IP address. This option is required if the peer tunnel endpoint is behind a NAT device. The verify-path ICMP request is sent to this IP address so that the peer can generate an ICMP response.

packet-size *bytes*

(Optional) The size of the packet that is used to verify an IPsec datapath before the st0 interface is brought up.

The packet size must be lower than the path maximum transmission unit (PMTU) minus tunnel overhead. The packet used for IPsec datapath verification must not be fragmented.

- **Range:** 64 to 1350 bytes
- **Default:** 64 bytes

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 15.1X49-D70.

`packet-size` option added in Junos OS Release 15.1X49-D120.

RELATED DOCUMENTATION

vpn-monitor | 1669
vpn (Security) | 1662

vpn (Security)

IN THIS SECTION

Syntax

```
vpn vpn-name {
    bind-interface interface-name;
    df-bit (clear | copy | set);
    distribution-profile (default-spc2-profile | default-spc3-profile | distribution-profile-name);
    copy-outer-dscp;
    establish-tunnels (immediately | on-traffic | responder-only | responder-only-no-rekey);
    match-direction (input | output);
    passive-mode-tunneling;
    tunnel-mtu tunnel-mtu;
    udp-encapsulate <dest-port dest-port>;
    ike {
        anti-replay-window-size anti-replay-window-size;
        gateway gateway-name;
        idle-time seconds;
        install-interval seconds;
        ipsec-policy ipsec-policy-name;
        no-anti-replay;
        proxy-identity {
            local ip-prefix;
            remote ip-prefix;
            service (any | service-name);
        }
    }
    manual {
        authentication {
            algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha1-96);
            key (ascii-text key | hexadecimal key);
        }
        encryption {
            algorithm (3des-cbc | aes-128-cbc | aes-128-gcm | aes-192-cbc | aes-256-cbc |
aes-256-gcm | des-cbc);
            key (ascii-text key | hexadecimal key);
        }
        external-interface external-interface-name;
        gateway ip-address;
        protocol (ah | esp);
        spi spi-value;
    }
        multi-sa {
            forwarding-class (expedited-forwarding | assured-forwarding | best-effort | network-
control);
        }
    traffic-selector traffic-selector-name {
        local-ip ip-address/netmask;
        remote-ip ip-address/netmask;

        protocol protocol_name/protocol_id;
        source-port low-high;
        destination-port low-high;
        metric metric_value;
        description description_value;
        term term_name {
        local-ip ip-address/netmask;
        remote-ip ip-address/netmask;
        protocol protocol_name/protocol_id;
        source-port low-high;
        destination-port low-high;
                                    }
    }
    vpn-monitor {
        destination-ip ip-address;
        optimized;
        source-interface interface-name;
        verify-path {
            destination-ip ip-address;
            packet-size bytes;
        }
    }
}
```

Hierarchy Level

```
[edit security ipsec]
```

Description

Configure an IPsec VPN. A VPN provides a means by which remote computers communicate securely across a public WAN suchas the Internet. A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The trafficthat flows between these two points passes through shared resources such as routers, switches, and othernetwork equipment that make up the public WAN. To secure VPN communication while passing throughthe WAN, the two participants create an IP Security (IPsec) tunnel. IPsec is a suite of related protocols for cryptographically securing communications at the IP Packet Layer.

Options

vpn-name Name of the VPN.

bind-interface Configure the tunnel interface to which the route-based virtual private network (VPN) is bound.

copy-outer-dscp Enable copying of Differentiated Services Code Point (DSCP) (outer DSCP+ECN) field from the outer IP header encrypted packet to the inner IP header plain text message on the decryption path. The benefit in enabling this feature is that after IPsec decryption, clear text packets can follow the inner CoS (DSCP+ECN) rules.

distribution-profile Specify a distribution-profile to distribute tunnels. The `distribution-profile` option is introduced to give the administrator an option to select which PICs in the chassis should handle tunnels associated with a certain VPN object. If the default profiles such as `default-spc3-profile` or `default-spc2-profile` are not selected, a new user-defined profile can be selected. In a profile, you need to mention the Flexible PIC Concentrator (FPC) slot and the PIC number. When such a profile is associated with a VPN object, all matching tunnels are distributed across these PIC's.

- Values:

    - default-spc2-profile—Default group for distributing tunnels on SPC2 only
    - default-spc3-profile—Default group for distributing tunnels on SPC3 only
    - distribution-profile-name—Name of the distribution profile.

df-bit

Specify how the device handles the Don't Fragment (DF) bit in the outer header.

On SRX5400, SRX5600, and SRX5800 devices, the DF-bit configuration for VPN only works if the original packet size is smaller than the st0 interface MTU, and larger than the external interface-ipsec overhead.

- Values:
    - `clear`—Clear (disable) the DF bit from the outer header. This is the default.
    - `copy`—Copy the DF bit to the outer header.
    - `set`—Set (enable) the DF bit in the outer header.

establish-tunnels

Specify when IKE is activated: immediately after VPN information is configured and configuration changes are committed, or only when data traffic flows. If this configuration is not specified, IKE is activated only when data traffic flows.

- Values:
    - `immediately`—IKE is activated immediately after VPN configuration changes are committed.

      Starting with Junos OS Release 15.1X49-D70, a warning message is displayed if you configure the `establish-tunnels immediately` option for an IKE gateway with `group-ike-id` or `shared-ike-id` IKE user types (for example, with AutoVPN or a remote access VPN). The `establish-tunnels immediately` option is not appropriate for these VPNs because multiple VPN tunnels may be associated with a single VPN configuration. Committing the configuration will succeed, however the `establish-tunnels immediately` configuration is ignored. The state of the tunnel interface will be up all the time, which was not the case in previous releases when the `establish-tunnels immediately` option was configured.
    - `on-traffic`—IKE is activated only when data traffic flows and must to be negotiated with the peer gateway. This is the default behavior.

  - `responder-only`—Responds to IKE negotiations that are initiated by the peer gateway, but does not initiate IKE negotiations from the device. This option is required when another vendor's peer gateway expects the protocol and port values in the traffic selector from the initiating gateway. `responder-only` option added in Junos OS Release 19.1R1.

    This option is supported on unified iked process that is not enabled by default. Administrators must execute the `request system software add optional://junos-ike.tgz` command to load the `junos-ike` package.

  - `responder-only-no-rekey`—Option does not establish any VPN tunnel from the device, so the VPN tunnel is initiated from the remote peer. An established tunnel does not start any rekeying from the device and relies on the remote peer to initiate this rekeying. If rekeying does not occur, then the tunnel is brought down after hard-lifetime expires.

    This option is supported on unified iked process that is not enabled by default. Administrators must execute the `request system software add optional://junos-ike.tgz` command to load the `junos-ike` package.

ike

Define an IKE-keyed IPsec VPN.

manual

Define a manual IPsec security association (SA).

multi-sa

Negotiate multiple security association (SAs) based on configuration choice. Multiple SAs negotiates with the same traffic selector on the same IKE SA.

traffic-selector

Configure multiple sets of local IP address prefix, remote IP address prefix, source port range, destination port range, and protocol as a traffic selector for an IPsec tunnel.

match-direction

Direction for which the rule match is applied

- Values:

  - input—Match on input to interface

  - output—Match on output from interface

passive-mode-tunneling

No active IP packet checks before IPSec encapsulation

tunnel-mtu

Maximum transmit packet size

- **Range:** 256 through 9192

udp-encapsulation

(Optional) Use the specified UDP destination port for the UDP header that is appended to the ESP encapsulation. Enable multiple path forwarding of IPsec traffic by adding a UDP header to the IPsec encapsulation of packets. Doing this increases the throughput of IPsec traffic. If you do not enable UDP encapsulation, all the IPsec traffic follows a single forward path rather than using multiple available paths.

- **Range:** 1025 through 65536. Do not use 4500.
- **Default:** If you do not include the udp-dest-port statement, the default UDP destination port is 4565.

vpn-monitor

Configure settings for VPN monitoring.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5.

Support for IPv6 addresses added in Junos OS Release 11.1.

Support for `copy-outer-dscp` added in Junos OS Release 15.1X49-D30.

`verify-path` keyword and `destination-ip` added in Junos OS Release 15.1X49-D70.

`packet-size` option added in Junos OS Release 15.1X49-D120.

Support for `term`, `protocol`, `source-port`, `destination-port`, `metric`, and `description` options introduced in Junos OS Release 21.1R1.

RELATED DOCUMENTATION

IPsec Overview | 20

vpn-monitor

IN THIS SECTION

Syntax

```
vpn-monitor {
    destination-ip ip-address;
    optimized;
    source-interface interface-name;
    verify-path {
        destination-ip ip-address;
        packet-size bytes;
    }
}
```

Hierarchy Level

```
[edit security ipsec vpn vpn-name]
```

Description

Configure settings for VPN monitoring.

Options

destination-ip Specify the destination of the Internet Control Message Protocol (ICMP) pings. If this statement is used, the device uses the peer's gateway address by default.

optimized Specify that VPN monitoring optimization is enabled for the VPN object. When VPN monitoring optimization is enabled, the SRX Series Firewall only sends ICMP echo requests (pings) when there is outgoing traffic and no incoming traffic from the configured peer through the VPN tunnel. If there is incoming traffic through the VPN tunnel, the SRX Series Firewall considers the tunnel to be active and does not send pings to the peer.

Because ICMP echo requests are only sent when needed to determine peer liveliness, VPN monitoring optimization can save resources on the SRX Series Firewall. Also, ICMP echo requests can activate costly backup links that would otherwise not be used.

This option is disabled by default.

source-interface Specify the source interface for ICMP requests (VPN monitoring “hellos”). If no source interface is specified, the device automatically uses the local tunnel endpoint interface.

verification-path Specify the verification path to verify the IPsec datapath before the secure tunnel (st0) interface is activated and route(s) associated with the interface are installed in the Junos OS forwarding table.

- `destination-ip` *`ip-address`*—Original, untranslated IP address of the peer tunnel endpoint that is behind a NAT device. This IP address must not be the NAT translated IP address. This option is required if the peer tunnel endpoint is behind a NAT device. The verify-path ICMP request is sent to this IP address so that the peer can generate an ICMP response.

- `packet-size` *`bytes`*—(Optional) The size of the packet that is used to verify an IPsec datapath before the st0 interface is brought up. The packet size must be lower than the path maximum transmission unit (PMTU) minus tunnel overhead. The packet used for IPsec datapath verification must not be fragmented. The range of the packet size is 64 to 1350 bytes and the default packet size value is 64 bytes

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 8.5. `verify-path` keyword and `destination-ip` added in Junos OS Release 15.1X49-D70. `packet-size` option added in Junos OS Release 15.1X49-D120.

RELATED DOCUMENTATION

IPsec Overview | 20

vpn (Security) | 1662

windows-logon (Juniper Secure Connect)

IN THIS SECTION

Syntax

```
windows-logon {
    auto-dialog-open;
    disconnect-at-logoff;
    domain domain;
    eap-auth;
    flush-credential-at-logoff;
    lead-time-duration seconds;
    mode (automatic | manual);
}
```

Hierarchy Level

```
[edit security remote-access client-config]
```

Description

Define windows logon settings for the Juniper Secure Connect remote client device.

Options

auto-dialog-open Automatically open dialog for connection establishment.

disconnect-at-logoff Disconnect the session after logoff.

domain Domain name for automatic windows logon.

eap-auth EAP authentication method before the profile selection.

flush-credential-at-logoff Flush cached credentials after logoff.

lead-time-duration Lead time duration for domain logon in seconds.

- **Default:** 45 seconds
- **Range:** 0 through 120

mode

Set windows logon mode.

- Values:
  - automatic—Automatic Windows logon with configured credentials.
  - manual—Manual Windows logon.

Required Privilege Level

security

Release Information

Statement introduced in Junos OS Release 20.3R1.

RELATED DOCUMENTATION

Juniper Secure Connect Administrator Guide

xauth-attributes

IN THIS SECTION

Syntax (inet)

```
xauth-attributes {
    primary-dns IP address;
    primary-wins IP address;
    secondary-dns IP address;
    secondary-wins IP address;
}
```

Syntax (inet6)

```
xauth-attributes {
    primary-dns-ipv6 IP address;
    secondary-dns-ipv6  IP address;
}
```

Hierarchy Level

```
[edit access address-assignment pool <name> family inet]
```

Hierarchy Level (inet6)

```
[edit access address-assignment pool <name> family inet6]
```

Description

Configure XAuth attributes to use in XAuth authentication.

Options

- `apply-groups`—Groups from which to inherit configuration data.
- `apply-groups-except`—Do not inherit configuration data from these groups.
- `primary-dns`—Specify the primary-dns IP address.
- `secondary-dns`—Specify the secondary-dns IP address.
- `primary-wins`—Specify the primary-wins IP address.
- `secondary-wins`—Specify the secondary-wins IP address.
- `primary-dns-ipv6`—Specify the primary-dns IPv6 address.
- `secondary-dns-ipv6`—Specify the secondary-dns IPv6.

Required Privilege Level

security—To view this statement in the configuration.

security-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 10.4.

`xauth-attributes` option under `inet6` is introduced in Junos OS Release 20.3R1.

19

CHAPTER

Operational Commands

clear security group-vpn member group

IN THIS SECTION

Syntax

```
clear security group-vpn member group <vpn vpn-name> <group-id group-id>
```

Description

Clear all current information for IKE, TEK, and KEK SAs. Group VPNv2 is supported on MX Series routers, SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewalls.

Options

none	Clear SA information for all groups.
vpn *vpn-name*	(Optional) Clear SA information for the specified VPN name.
group-id *group-id*	(Optional) Clear SA information for the specified group identifier.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

Group VPNv2 Overview

clear security group-vpn member ike security-associations

IN THIS SECTION

Syntax

```
clear security group-vpn member ike security-associations [index SA-index] [peer-ipaddress]
```

Description

Clear IKE security association (SA) for a group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewalls.

Options

- none—Clear all IKE SAs for the group member.
- `index`—(Optional) Clear the IKE SA with this index number.
- *peer-ipaddress*—(Optional) Clear the IKE SA with this peer.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

show security group-vpn member ike security-associations | 1782

Group VPNv2 Overview

clear security group-vpn member ipsec security-associations

IN THIS SECTION

Syntax

```
clear security group-vpn member ipsec security-associations [index SA-index]
```

Description

Clear group VPN SA for a group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewalls.

Options

- none—Clear all group VPN SAs for the group member.
- `index`—(Optional) Clear the group VPN SA with this index number.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

show security group-vpn member ipsec security-associations | 1792
Group VPNv2 Overview | 758

clear security group-vpn member ipsec security-associations statistics

IN THIS SECTION

Syntax

```
clear security group-vpn member ipsec security-associations statistics <group-id group-id>
```

Description

Clear IPsec SA statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewalls.

Options

none	Clear IPsec SA statistics for all groups.
group-id *group-id*	(Optional) Clear IPsec SA statistics for the specified group identifier.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

clear security group-vpn member ipsec statistics

IN THIS SECTION

Syntax

```
clear security group-vpn member ipsec statistics <index index>
```

Description

Clear IPsec statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewalls.

Options

none — Clear IPsec statistics for all groups.

index *index* — (Optional) Clear the IPsec statistics for the SA with this index number.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

clear security group-vpn server

IN THIS SECTION

Syntax

```
clear security group-vpn server [group group-name | group-id group-id] [now]
```

Description

Clear active members for a specified group. If no options are specified, members are cleared from all groups. After this command is issued, members will need to reregister. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewalls.

An IKE SA can be used by a group member to register to multiple groups. When you clear members for a specified group, all existing IKE SAs that could be used to register to the group are also cleared.

Options

- none—All members are cleared from all groups.
- `group`—(Optional) Clear members and SAs for the specified group name.
- `group-id`—(Optional) Clear members and SAs for the specified group identifier.
- `now`—(Optional) Immediately clear all group-related information.

Required Privilege Level

clear

Output Fields

If there is a problem with the command, one of the following messages appears:

- Group does not exist
- Group is in the process of deletion
- Error in clear members
- Warning Message; Fail to push delete to members as server-member-communication is not configured.

RELATED DOCUMENTATION

show security group-vpn server registered-members | 1828
Group VPNv2 Overview | 758

clear security group-vpn server server-cluster statistics

IN THIS SECTION

Syntax

```
clear security group-vpn server server-cluster statistics <group group-name> <group-id group-id>
```

Description

Clear Group VPNv2 server cluster statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewalls.

Options

none — Clear Group VPNv2 server cluster statistics for all groups.

group *group-name* — (Optional) Clear Group VPNv2 server cluster statistics for the specified group name.

group-id *group-id*	(Optional) Clear Group VPNv2 server cluster statistics for the specified group identifier.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758
Understanding Group VPNv2 Server Clusters | 818

clear security group-vpn server statistics

IN THIS SECTION

Syntax

```
clear security group-vpn server statistics <group group-name> <group-id group-id>
```

Description

Clear group statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewalls.

Options

none	Clear statistics for all groups.
group *group-name*	(Optional) Clear statistics for the specified group name.
group-id *group-id*	(Optional) Clear statistics for the specified group identifier.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

show security group-vpn server statistics | 1836

Group VPNv2 Overview | 758

clear security ike active-peer aaa-username

IN THIS SECTION

Syntax

```
clear security ike active-peer aaa-username username
```

Description

Clears all IKE security association (SA) of a user.

Options

- none—Clear all IKE SAs for the user.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 22.3R1.

clear security ike respond-bad-spi-count

IN THIS SECTION

Syntax

```
clear security ike respond-bad-spi-count <gateway-name>
```

Description

Clear information about invalid Internet Key Exchange (IKE) security parameter index (SPI) counters.

Options

- none—Clear all invalid SPI counters.
- *gateway-name* —(Optional) Clear the invalid SPI counters for the given gateway.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 8.5.

RELATED DOCUMENTATION

ike (Security) | **1524**

clear security ike security-associations

IN THIS SECTION

Syntax

```
clear security ike security-associations
<peer-address>
<family (inet  | inet6)>
<fpc slot-number>
<index SA-index-number>
<kmd-instance (all | kmd-instance-name)>
<pic slot-number>
<node-local>
<port port-number>
<sa-type shortcut>
<srg-id id-number>
<ha-link-encryption>
```

Description

Clear information about the current Internet Key Exchange security associations (IKE SAs). For IKEv2, the device clears the information about the IKE SAs and the associated IPSec SA.

Options

- none—Clear all IKE SAs.
- `peer-address` —(Optional) Clear IKE SAs for the destination peer at this IP address.
- `family`—(Optional) Clear IKE SAs by family.
  - `inet`—IPv4 address family.
  - `inet6`—IPv6 address family.
- `fpc` *`slot-number`* —Specific to SRX Series Firewalls. Clear information about existing IKE SAs in this Flexible PIC Concentrator (FPC) slot.
- `index` *`SA-index-number`* —(Optional) Clear the IKE SA with this index number.
- `kmd-instance`—Clear information about existing IKE SAs in the key management process (the daemon, which in this case is KMD) identified by FPC *`slot-number`* and PIC *`slot-number`*. Specific to SRX Series Firewalls.
  - `all`—All KMD instances running on the Services Processing Unit (SPU).
  - *`kmd-instance-name`*—Name of the KMD instance running on the SPU.
- `node-local`—(Optional) Clear information about IKE SAs for node-local tunnels in a Multinode High Availability setup.
- `pic` *`slot-number`* —Clear information about existing IKE SAs in this PIC slot. Specific to SRX Series Firewalls.
- `port` *`port-number`*—(Optional) Port number of SA (1 through 65,535).
- `sa-type shortcut`—(Optional for ADVPN) Type of SA. `shortcut` is the only option for this release.
- `ha-link-encryption`—(Optional) Clear information about the current IKE SAs for high availability (HA) link tunnel only. When you enable High Availability feature, you cannot delete customer tunnels on the backup node.
- `srg-id`—(Optional) Clear information related to a specific services redundancy group (SRG) in a Multinode High Availability setup.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 8.5. The `fpc`, `pic`, and `kmd-instance` options added in Junos OS Release 9.3. The `port` option added in Junos OS Release 10.0. The `family` option added in Junos OS Release 11.1.

Support for the `ha-link-encryption` option added in Junos OS Release 20.4R1.

Support for the `srg-id` option added in Junos OS Release 22.4R1.

Support for the `node-local` option added in Junos OS Release 23.2R1.

RELATED DOCUMENTATION

https://www.juniper.net/documentation/us/en/software/junos/vpn-ipsec/topics/ref/command/show-security-ike-security-associations.html

clear security ipsec security-associations

IN THIS SECTION

Syntax

```
clear security ipsec security-associations
<family (inet  | inet6)>
<fpc slot-number>
<index SA-index-number>
<kmd-instance (all | kmd-instance-name)>
<node-local>
<pic slot-number>
<srg-id id-number>
<ha-link-encryption>
```

Description

Clear information about IPsec security associations (SAs).

Options

- none—Clear all IPsec SAs.
- `family`—(Optional) Clear SAs by family.
  - `inet`—IPv4 address family.
  - `inet6`—IPv6 address family.
- `fpc` *`slot-number`* —Clear information about existing IPsec SAs in this Flexible PIC Concentrator (FPC) slot. Specific to SRX Series Firewalls.
- `index` *`SA-index-number`* —(Optional) Clear the IPsec SA with this index number.
- `kmd-instance`—Clear information about existing IPsec SAs in the key management process (the daemon, which in this case is KMD) identified by FPC *`slot-number`* and PIC *`slot-number`* Specific to SRX Series Firewalls.
  - `all`—All KMD instances running on the Services Processing Unit (SPU).
  - *`kmd-instance-name`*—Name of the KMD instance running on the SPU.

- `node-local`—(Optional) Clear information about IPsec SAs for node-local tunnels in a Multinode High Availability setup.
- `pic` *`slot-number`* —Clear information about existing IPsec SAs in this PIC slot. Specific to SRX Series Firewalls.
- `ha-link-encryption`—(Optional) Clear information about IPsec SAs for interchassis link tunnel only. See "ipsec (High Availability)" on page 1544. When you enable High Availability feature, you cannot delete customer tunnels on the backup node.
- `srg-id`—(Optional) Clear statistics related to a specific services redundancy group (SRG) in a Multinode High Availability setup..

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 8.5. The `fpc`, `pic`, and `kmd-instance` options added in Junos OS Release 9.3. The `family` option added in Junos OS Release 11.1.

Support for the `ha-link-encryption` option added in Junos OS Release 20.4R1.

Support for the `srg-id` option added in Junos OS Release 22.4R1.

Support for the `node-local` option added in Junos OS Release 23.2R1.

RELATED DOCUMENTATION

show security ipsec security-associations | **1899**

clear security ipsec statistics

IN THIS SECTION

Syntax

```
clear security ike statistics
<fpc slot-number>
<index SA-index-number>
<kmd-instance (all |            kmd-instance-name        )>
<pic slot-number>
<srg-id id-number>
```

Description

Clear IPsec statistics on the device.

Options

- none—Clear all IPsec statistics.
- `fpc` *`slot-number`* —Specific to SRX Series Firewalls. Clear statistics about existing IPsec security associations (SAs) in this Flexible PIC Concentrator (FPC) slot.

- `index` *`SA-index-number`* —(Optional) Clear the IPsec statistics for the SA with this index number.
- `kmd-instance`—Specific to SRX Series Firewalls. Clear information about existing IKE SAs in the key management process (the daemon, which in this case is KMD) identified by FPC *`slot-number`* and PIC *`slot-number`* .
  - `all`—All KMD instances running on the Services Processing Unit (SPU).
  - *`kmd-instance-name`*—Name of the KMD instance running on the SPU.
- `pic` *`slot-number`* —Specific to SRX Series Firewalls. Clear statistics about existing IPsec SAs in this PIC slot.
- `srg-id` *`id-number`* —Clear statistics related to a specific services redundancy group (SRG) in a Multinode High Availability setup.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 8.5. `fpc and pic` options added in Junos OS Release 9.3. `kmd-instance` option added in Junos OS Release 10.4.

`srg-id` option added in Junos OS Release 22.4R1.

RELATED DOCUMENTATION

show security ipsec statistics | 1938

clear security ike stats

IN THIS SECTION

Syntax

```
clear security ike stats
```

Description

Clears the global IKE statistics.

Required Privilege Level

clear

Sample Output

clear security ike stats

```
user@host> clear security ike stats
```

command-name

The `clear security ike stats` command does not display any output. To view the IKE statistics, run the `show security ike stats detail` command.

show security ike stats detail

```
user@host> show security ike stats detail
Total IKE SA and Tunnel Count Statistics:
  Number of IKE SAs: 2             Number of IPsec Tunnels: 2

IKE_SA_INIT exchange stats:
 Initiator stats:                                  Responder stats:
  Request Out             : 0                       Request In               : 0
  Response In             : 0                       Response Out             : 0
  Invalid KE Payload In   : 0                       Invalid KE Payload Out   : 0
  No Proposal Chosen In   : 0                       No Proposal Chosen Out   : 0
  Cookie Request In       : 0                       Cookie Request Out       : 0
  Cookie Response Out     : 0                       Cookie Response In       : 0
  Res Invalid IKE SPI     : 0                       Res DH Gen Key Fail      : 0
  Res Verify SA Fail      : 0                       Res Invalid DH Group Conf: 0
  Res IKE SA Fill Fail    : 0                       Res Get CAs Fail         : 0
  Res Verify DH Group Fail: 0                       Res Get VID Fail         : 0
  Res DH Compute Key Fail : 0                       Res DH Compute Key Fail  : 0

IKE_AUTH exchange stats:
 Initiator stats:                                  Responder stats:
  Request Out             : 0                       Request In               : 0
  Response In             : 0                       Response Out             : 0
  No Proposal Chosen In   : 0                       No Proposal Chosen Out   : 0
  TS Unacceptable In      : 0                       TS Unacceptable Out      : 0
  Authentication Failed In: 0                       Authentication Failed Out: 0

IKE SA Rekey CREATE_CHILD_SA exchange stats:
 Initiator stats:                                  Responder stats:
  Request Out              : 0                       Request In             : 0
  Response In              : 0                       Response Out           : 0
  No Proposal Chosen In    : 0                       No Proposal Chosen Out : 0
  Invalid KE In            : 0                       Invalid KE Out         : 0
  Res DH Compute Key Fail : 0                        Res DH Compute Key Fail: 0
  Res Verify SA Fail       : 0
  Res Fill IKE SA Fail     : 0
  Res Verify DH Group Fail: 0

IPsec SA Rekey CREATE_CHILD_SA exchange stats:
 Initiator stats:                                  Responder stats:
  Request Out              : 0                       Request In             : 0
  Response In              : 0                       Response Out           : 0
  No Proposal Chosen In    : 0                       No Proposal Chosen Out : 0
  Invalid KE In            : 0                       Invalid KE Out         : 0
  TS Unacceptable In       : 0                       TS Unacceptable Out    : 0
  Res DH Compute Key Fail : 0                        Res DH Compute Key Fail: 0
  Res Verify SA Fail       : 0
  Res Verify DH Group Fail: 0
  Res Verify TS Fail       : 0

Total IKE message failure stats:
  Discarded            : 0                           ID error      : 0
  Integrity fail       : 0                           Invalid SPI   : 0
  Invalid exchange type: 0                           Invalid length: 0
  Disorder             : 0
```

Release Information

Command is introduced in Junos OS Release 20.1R1.

RELATED DOCUMENTATION

Configure the Certificate Expiration Trap

Enable Peer Down and IPsec Tunnel Down Traps

show security ipsec statistics | 1938

clear security ipsec tunnel-events-statistics

IN THIS SECTION

Syntax

```
clear security ipsec tunnel-events-statistics
```

Description

Clear IPsec tunnel event statistics.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 12.3X48-D10.

RELATED DOCUMENTATION

show security ipsec tunnel-events-statistics | 1955

clear security pki key-pair (Local Certificate)

IN THIS SECTION

Syntax

```
clear security pki key-pair (all | certificate-id          certificate-id       )
```

Description

Clear public key infrastructure (PKI) key pair information for local digital certificates on the device.

Options

- `all`—Clear key pair information for all local certificates.
- `certificate-id` *`certificate-id`* —Clear key pair information for the local certificate with this certificate ID.

Required Privilege Level

clear and security

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 8.5.

RELATED DOCUMENTATION

show security pki certificate-request (View) | 1965

clear security pki local-certificate (Device)

IN THIS SECTION

Syntax

```
clear security pki local-certificate (all | certificate-id          certificate-id         |
system-generated)
```

Description

Clear public key infrastructure (PKI) information for local digital certificates on the device.

Options

- `all`—Clear information for all the local digital certificates on the device.

  You cannot clear the automatically generated self-signed certificate using `clear security pki local-certificate all` command. To clear the self-signed certificate you need to use `system-generated` as an option.

- `certificate-id` *`certificate-id`* —Clear the specified local digital certificate with this certificate ID.

- `system-generated`—Clear the existing automatically generated self-signed certificate and generate a new self-signed certificate.

Required Privilege Level

clear and security

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear security pki local-certificate all

```
user@host> clear security pki local-certificate all
```

Sample Output

clear security pki local-certificate system-generated

```
user@host> clear security pki local-certificate system-generated
```

Release Information

Command modified in Junos OS Release 9.1.

Starting in Junos OS Release 20.1R1 on vSRX Virtual Firewall 3.0, you can safeguard the private keys used by PKID and IKED using Microsoft Azure Key Vault hardware security module (HSM) service. You can establish a PKI based VPN tunnel using the keypairs generated at the HSM. The hub `certificate-id` option under certificate-id is not available for configuration after generating HSM key-pair.

Starting in Junos OS Release 20.4R1 on vSRX Virtual Firewall 3.0, you can safeguard the private keys used by PKID and IKED using AWS Key Management Service (KMS). You can establish a PKI based VPN tunnel using the keypairs generated by the KMS. The hub `certificate-id` option under certificate-id is not available for configuration after generating PKI key-pair.

> **NOTE**: You cannot manually re-enroll the local certificates when you re-generate key-pairs, if you are not generating key-pairs during re-enrollment. A warning **HSM does not support auto re-enrollment with new keypair error: configuration check-out failed** is displayed in the output of the `show security pki auto-re-enrollment` command.
>
> Also, when you clear the local certificates using the `run clear security pki local-certificate all` and `run clear security pki key-pair all` commands you will receive a warning **Key pair deleted successfully but still present at HSM. Please purge the keypair from keyvault before re-using the name.**

RELATED DOCUMENTATION

show security pki local-certificate (View) | 1972

request security pki local-certificate generate-self-signed (Security) | 1754

clear security pki statistics

IN THIS SECTION

Syntax

```
clear security pki statistics
```

Description

Clear PKI statistics on the device.

Options

None

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 21.4R1.

RELATED DOCUMENTATION

show security pki statistics | 1983

request security ike debug-disable

IN THIS SECTION

Syntax

```
request security ike debug-disable
```

Description

Disable IKE debugging.

Required Privilege Level

maintenance

Output Fields

This command produces no output.

Release Information

Command introduced in Release Junos OS 11.4R3.

RELATED DOCUMENTATION

request security ike debug-enable | 1714

show security ike debug-status | 1847

request security ike debug-enable

IN THIS SECTION

Syntax

```
request security ike debug-enable local local-ip-address remote remote-ip-address
```

Description

Enable IKE tracing on a single VPN tunnel specified by a local and a remote IP address. Use of this command is an alternative to configuring IKE traceoptions; you do not require any configuration to use this command. This command only traces a single tunnel, whereas configuring IKE traceoptions affects all VPN tunnels on the SRX Series Firewalls.

NOTE: SRX Series Firewalls and MX-SPC3 Services Card supports this command. MX Series device with Multiservices Modular Interfaces Card (MS-MIC) or Multiservices Modular PIC Concentrator (MS-MPC) does not support this command.

To use this command:

1. Identify the local and remote IP addresses of the VPN tunnel you want to trace.

2. Enable IKE tracing on the VPN tunnel with this command.

3. Attempt tunnel establishment to capture trace information to the log file:

   - For the SRX Series Firewalls and vSRX Virtual Firewall running kmd process, the trace information is stored in `/var/log/kmd` file.

   - For the MX-SPC3 Services Card, SRX Series Firewalls and vSRX Virtual Firewall running iked process (including mixed mode), the trace information is stored in `/var/log/iked` file.

   If you've configured to save the trace messages into a specific file under the `[edit security ike traceoptions]` hierarchy level, the trace information is stored in the specified file name.

4. Disable per-tunnel IKE tracing with the **request security ike debug-disable** command.

5. Review the log file with the following command:

   - For the SRX Series Firewalls and vSRX Virtual Firewall running kmd process, execute the `show log kmd` or the file name specified under the `[edit security ike traceoptions]` hierarchy level.

   - For the MX-SPC3 Services Card, SRX Series Firewalls and vSRX Virtual Firewall running iked process (including mixed mode), execute the `show log iked` or the file name specified under the `[edit security ike traceoptions]` hierarchy level.

You can use the **show security ike debug-status** command:

- to view the status of the per-tunnel IKE tracing operation.

- to view the status of the interchassis link tunnel only.

Options

- `local` *`local-ip-address`*—The address of the local VPN peer.

- `remote` *`remote-ip-address`*—The address of the remote VPN peer.

Required Privilege Level

maintenance

Release Information

Command introduced in Junos OS Release 11.4R3.

RELATED DOCUMENTATION

request security ike debug-disable | 1713

show security ike debug-status | 1847

clear security tcp-encap statistics

IN THIS SECTION

Syntax

```
clear security tcp-encap statistics
```

Description

Clear TCP encapsulation statistics.

Required Privilege Level

clear

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 15.1X49-D80.

RELATED DOCUMENTATION

show security tcp-encap statistics | 2005

request security pki ca-certificate ca-profile-group default-trusted-ca-certs

IN THIS SECTION

Syntax

```
request security pki ca-certificate ca-profile-group default-trusted-ca-certs download [check-
server | no-forwarding | status]
```

Description

When you setup dynamic update of trusted CA bundle, you use this command to -

- Explicitly instruct the SRX Series Firewall or a Junos OS device to manually download the default trusted CA certificates from CDN server. It checks the available version on the CDN server and downloads it.
- Check connectivity to the CDN server for the Junos OS device.
- Monitor the status of default trusted CA certificates downloaded from CDN server.

Options

download Download default trusted CA certificates from a CDN server.

Use this option when you need to explicitly download default trusted CA certificates in addition to periodic download.

check-server

Check connectivity to CDN server to download default trusted CA certificates.

This command downloads the manifest file and displays the trusted-ca-bundle version available in CDN server

no-forwarding

No forwarding.

status

Check current status of default trusted CA certificates downloaded from CDN server.

This option displays the default trusted CA certificates version number and version date.

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki ca-certificate ca-profile-group default-trusted-ca-certs download

- Success scenario

```
user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
download
Connection to CDN server is successful. Default trusted CA certs bundle version available is
   <version> dated <date>
   Downloading the trusted CA certs bundle...
   Download successful
   Updating trusted CA certs to default CA profile group <ca-profile-group>
   Added <num> CAs and removed <num> CAs from default trusted CA profile group test
```

- Failure scenario - 1

  ```
  user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
  download
  Connection to CDN server is unsuccessful
  ```

- Failure scenario - 2

  ```
  user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
  download
  Connection to CDN server is successful. Default trusted CA certs bundle
     version available is <version_no> dated <date>.
     Downloading the trusted CA certs...
     Download Failed.
  ```

Sample Output

request security pki ca-certificate ca-profile-group default-trusted-ca-certs download check-server

```
user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
download check-server
Connection to CDN server is successful. Default trusted CA certs bundle version available is
<version_no> dated <date>.
```

Sample Output

request security pki ca-certificate ca-profile-group default-trusted-ca-certs download status

- Success scenario

```
user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
download status
Default trusted CA certs bundle available on device is <version_no> dated <date>.
```

- Failure scenario

```
user@host> request security pki ca-certificate ca-profile-group default-trusted-ca-certs
download status
Default trusted CA certs bundle is not available.
```

Release Information

Statement introduced in Junos OS Release 23.2R1.

RELATED DOCUMENTATION

default-trusted-ca-certs (Security)

Dynamic Update of Trusted CA Bundle

request security pki ca-certificate ca-profile-group load

IN THIS SECTION

Syntax

```
request security pki ca-certificate ca-profile-group load ca-group-name ca-group-name filename
[path/filename | default]
```

Description

For SSL forward proxy, you need to load trusted CA certificates on your system. By default, Junos OS provides a list of trusted CA certificates that include default certificates used by common browsers. Alternatively, you can define your own list of trusted CA certificates and import them on to your system.

Use this command to load the default certificates or to specify a path and filename of trusted CA certificates that you define.

The `default` option is not supported on PTX10003-80C, PTX10003-160C, and PTX10008 routers.

Starting in Junos OS Release 21.4R1, you can get the status of CA certificates configured under default CA profile group by executing "request security pki ca-profile-group-status" on page 1734 command .

With "request security pki ca-profile-group-status" on page 1734 command, you can verify the number of CA certificates loaded and number of CA certificates missing within a CA profile group.

Starting in Junos OS Release 23.2R1, when you configure dynamic update of trusted CA bundle using the statement default-trusted-ca-certs (Security), the process of loading the default trusted CA certificates happens in the background. During this process, PKID response might slowdown for few minutes.

Options

ca-group-name *ca-group-name* — Load the specified CA group profile.

filename *path/filename* — Directory location and filename of the trusted CA certificates defined by you.

filename default — Load the trusted CA certificates available by default.

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki ca-certificate ca-profile-group load (default)

```
user@host> request security pki ca-certificate ca-profile-group load ca-group-name ca-default
filename default

Loading of certs started
Loading of <no-of-certs> trusted CA certs started in the background. PKID response might be slow
for next several minutes.
```

Sample Output

request security pki ca-certificate ca-profile-group load (path/filename)

```
user@host> request security pki ca-certificate ca-profile-group load ca-group-name ca-manual
filename /var/tmp/firefox-all.pem

Do you want to load this CA certificate ? [yes,no] (no) yes

Loading 196 certificates for group 'ca-manual'.
ca-manual_1_sysgen: Loading done.
ca-manual_2_sysgen: Loading done.
ca-manual_3_sysgen: Loading done.
ca-manual_4_sysgen: Loading done.
ca-manual_5_sysgen: Loading done.
ca-manual_6_sysgen: Loading done.

...
ca-manual_195_sysgen: Loading done.
ca-manual_196_sysgen: Loading done.
ca-profile-group 'ca-manual’ successfully loaded. Success[193] Skipped[3]
```

Release Information

Command introduced in Junos OS Release 12.1; `default` option added in Junos OS Release 12.1X47-D10.

RELATED DOCUMENTATION

show security pki ca-certificate

PKI Components In Junos OS | **33**

request security pki ca-profile-group-status | **1734**

request security pki ca-certificate enroll (Security)

IN THIS SECTION

Syntax

```
request security pki ca-certificate enroll ca-profile ca-profile-name
```

Description

Request a digital certificate from a certificate authority (CA) online by using the Simple Certificate Enrollment Protocol (SCEP).

Options

ca-profile *ca-profile-name*	CA profile name.

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki ca-certificate enroll

```
user@host> request security pki ca-certificate enroll ca-profile entrust
Received following certificates:
  Certificate: C=us, O=example, CN=First Officer
    Fingerprint: 46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f
  Certificate: C=us, O=example, CN=First Officer
    Fingerprint: bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17
  Certificate: C=us, O=example
    Fingerprint: 00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10
Do you want to load the above CA certificate ? [yes,no] (no) yes
```

Release Information

Command introduced in Junos OS Release 7.5.

RELATED DOCUMENTATION

show security pki ca-certificate (View) | **1959**

PKI Components In Junos OS | **33**

request security pki ca-certificate load (Security)

IN THIS SECTION

Syntax

```
request security pki ca-certificate load ca-profile ca-profile-name   filename path/
filename
```

Description

Manually load a certificate authority (CA) digital certificate from a specified location.

Options

ca-profile *ca-profile-name* Load the specified CA profile.

filename *path/filename* Directory location and filename of the CA digital certificate.

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki ca-certificate load

```
user@host> request security pki ca-certificate load ca-profile 2Kkey filename /var/tmp/
2Kkey.pem

Fingerprint:
  a0:08:bb:1f:75:96:76:cd:ee:db:36:10:b6:c6:d8:df:5e:02:05:05 (sha1)
  f5:58:6b:de:7c:d6:cd:90:5a:18:c3:0e:3d:95:da:25 (md5)
Do you want to load this CA certificate ? [yes,no] (no) yes

CA certificate for profile 2Kkey loaded successfully
```

Release Information

Command introduced in Junos OS Release 7.5.

RELATED DOCUMENTATION

show security pki ca-certificate

PKI Components In Junos OS | 33

request security pki ca-certificate verify (Security)

IN THIS SECTION

Syntax

```
request security pki ca-certificate verify ca-profile ca-profile-name
```

Description

Verify the digital certificate installed for the specified certificate authority (CA).

Options

`ca-profile` *`ca-profile-name`* —Display the specified CA profile.

Required Privilege Level

maintenance and security

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki ca-certificate verify ca-profile ca1 (CRL downloaded)

This user has downloaded the certificate revocation list (CRL).

```
user@host> request security pki ca-certificate verify ca-profile ca1
CA certificate ca1 verified successfully
```

request security pki ca-certificate verify ca-profile ca1 (CRL not downloaded)

This user has not downloaded the certificate revocation list (CRL).

```
user@host> request security pki ca-certificate verify ca-profile ca1
CA certificate ca1: CRL verification in progress. Please check the PKId debug logs for
completion status
```

request security pki ca-certificate verify ca-profile Root-CA (Verify enrolled CA certificate validity status on MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

You receive the following response when the CA certificate verification is failed. In this sample, the CA certificate verification is failed due to invalid CA certificate:

```
user@host> request security pki ca-certificate verify ca-profile Root-CA
CA certificate Root-CA verification failed. CA cert is not valid untill <05-19-2021 08:05>
```

request security pki ca-certificate verify ca-profile Root-CA (Verify enrolled CA certificate present in MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

You receive the following response when the CA certificate is missing:

```
user@host> request security pki ca-certificate verify ca-profile Root-CA
CA cert Root-CA Verification Failed. CA cert is missing
```

request security pki ca-certificate verify ca-profile CSO_37 (Verify local certificate status when the CA is unreachable for MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

You receive the following response when a CA is not reachable or CRL download has failed:

```
user@host> request security pki ca-certificate verify ca-profile CSO_37
CA certificate CSO_37 Verification Failed. Unreachable CA or CRL Download Failed
```

Release Information

Command introduced in Junos OS Release 8.5.

RELATED DOCUMENTATION

- ca-profile (Security PKI) | **1455**
- show security pki ca-certificate (View) | **1959**
- request security pki ca-profile-group-status (MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall) | **1735**
- PKI Components In Junos OS | **33**

request security pki crl load (Security)

IN THIS SECTION

Syntax

```
request security pki crl load ca-profile ca-profile-name  filename path/filename
```

Description

Manually install a certificate revocation list (CRL) on the device from a specified location.

Options

ca-profile *ca-profile-name*	Load the specified certificate authority (CA) profile.
filename *path/filename*	Directory location and filename of the CRL.

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki crl load

```
user@host> request security pki crl load ca-profile ca-test filename example-inter-
ca.crl
CRL for CA profile ca-test loaded successfully
```

Release Information

Command introduced in Junos OS Release 8.1.

RELATED DOCUMENTATION

PKI Components In Junos OS | 33

request security pki ca-profile-group-status

IN THIS SECTION

Syntax

```
request security pki ca-profile-group-status ca-group-name [ca-group-name|default]
```

Description

Get the status of CA certificates configured under default CA profile group. With this command, you can verify the number of CA certificates loaded and number of CA certificates missing within a CA profile group.

Options

ca-group-name *ca-group-name*	Load the specified CA group profile.
ca-group-name default	Load the trusted CA group profile available by default.

Required Privilege Level

maintenance

Output Fields

Table 125 on page 1735 lists the output fields for the `request security pki ca-profile-group-status` command.

Table 125: request security pki ca-profile-group-status Output Fields

Field Name	Field Description
No-of-Ca-Certs-Loaded	Total number of certificates successfully loaded for the specified CA profile group.
No-of-Ca-Certs-Load-Failure	Total number of certificates failed to load for the specified CA profile group.
Missing-Cert-id	Missing certificate Ids for the specified CA profile group.
Total-Certs	Total number of certificates imported for the specified CA profile group.

Sample Output

request security pki ca-profile-group-status (MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

```
user@host> request security pki ca-profile-group-status ca-group-name DEFAULT_CSO
No-of-Ca-Certs-Loaded:135 , No-of-Ca-Certs-Load-Failure: 0 , Missing-Cert-id: None ,Total-Certs:
135
```

Release Information

Command introduced in Junos OS Release 21.4R1.

RELATED DOCUMENTATION

show security ipsec statistics (MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall) | 1997

request security pki generate-certificate-request (Security)

IN THIS SECTION

Syntax

```
request security pki generate-certificate-request certificate-id certificate-id-name domain-name
domain-name   subject subject-distinguished-name
<add-ca-constraint>
<digest (sha1 | sha256)>
<email email-address>
```

```
<filename (path | terminal)>
<ip-address ip-address>
```

Description

Manually generate a local digital certificate request in the Public-Key Cryptography Standards #10 (PKCS-10) format.

Options

certificate-id *certificate-id-name*

Name of the local digital certificate and the public/private key pair.

domain-name *domain-name*

Fully qualified domain name (FQDN) provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.

subject *subject-distinguished-name*

Distinguished name format contains the following information:

- `DC`—Domain component
- `CN`—Common name
- `OU`—Organizational unit name
- `O`—Organization name
- `L`—Locality
- `ST`—State
- `C`—Country

digest

(Optional) Hash algorithm used to sign the certificate request.

- `sha1`—SHA-1 digests (default value for RSA or DSA only).
- `sha256`—SHA-256 digests for RSA or ECDSA only (default value for ECDSA).
- `sha-384`—SHA-384 digests for ECDSA only.

Starting in Junos OS Release 18.1R3, the default encryption algorithm that is used for validating automatically and manually generated self-signed PKI certificates is Secure Hash Algorithm 256 (SHA-256). Prior to Junos OS Release 18.1R3, SHA-1 is used as default encryption algorithm.

email *email-address* (Optional) E-mail address of the certificate holder.

filename (*path* | terminal) (Optional) Location where the local digital certificate request should be placed or the login terminal.

ip-address *ip-address* (Optional) IP address of the router.

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki generate-certificate-request

```
user@host> request security pki generate-certificate-request certificate-id local-entrust2
domain-name router2.example.net filename entrust-req2 subject cn=router2.example.net

Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIBoTCCAQoCAQAwGjEYMBYGA1UEAxMPdHAxLmp1bmlwZXIubmV0MIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQCiUFklQws1Ud+AqN5DDxRs2kVyKEhh9qoVFnz+
Hz4c9vsy3B8ElwTJlkmIt2cB3yifB6zePd+6WYpf57Crwre7YqPkiXM31F6z3YjX
H+1BPNbCxNWYvyrnSyVYDbFj8o0Xyqog8ACDfVL2JBWrPNBYy7imq/K9soDBbAs6
5hZqqwIDAQABoEcwRQYJKoZIhvcNAQkOMTgwNjAOBgNVHQ8BAf8EBAMCB4AwJAYD
VR0RAQH/BBowGIIWdHAxLmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQF
```

```
AAOBgQBc2rq1v5SOQXH7LCb/FdqAL8ZM6GoaN5d6cGwq4bB6a7UQFgtoH406gQ3G
3iH0Zfz4xMIBpJYuGd1dkqgvcDoH3AgTsLkfn7Wi3x5H2qeQVs9bvL4P5nvEZLND
EIMUHwteolZCiZ70fO9Fer9cXWHSQs1UtXtgPqQJy2xIeImLgw==
-----END CERTIFICATE REQUEST-----
Fingerprint:
0d:90:b8:d2:56:74:fc:84:59:62:b9:78:71:9c:e4:9c:54:ba:16:97 (sha1)
1b:08:d4:f7:90:f1:c4:39:08:c9:de:76:00:86:62:b8 (md5)
```

Release Information

Command introduced in Junos OS Release 7.5. Support for `digest` option added in Junos OS Release 12.1X45-D10.

RELATED DOCUMENTATION

show security pki certificate-request (View) | 1965

request security pki generate-key-pair (Security)

IN THIS SECTION

Syntax

```
request security pki generate-key-pair certificate-id certificate-id-name
<size (256 | 384 | 1024 | 2048 | 4096 | 521)>
<type (dsa | ecdsa | rsa)>
```

Description

Generate a public key infrastructure (PKI) public/private key pair for a local digital certificate.

Options

certificate-id *certificate-id-name*

Name of the local digital certificate and the public/private key pair.

size

Key pair size. The key pair size can be 256, 384, 521, 1024, 2048, or 4096 bits. Key pair sizes of 256, 384, and 521 bits are compatible with ECDSA. For Digital Signal Algorithm (DSA) and Rivest Shamir Adleman (RSA), algorithms the size must be 1024, 2048, or 4096. The default key pair size is 1024 for DSA and 2048 for RSA.

The following are supported when ECDSA-521 signatures are used:

- Load a complete certificate, which is generated using an external tool like OpenSSL into PKI.
- Manually generate a Certificate Signing Request (CSR) for a local certificate and sending the CSR to a (Certificate Authority) CA server to enroll.
- Automatic enroll with CA server.

type

The algorithm to be used for encrypting the public/private key pair:

- `ecdsa`—ECDSA encryption
- `dsa`— DSA encryption
- `rsa`—RSA encryption (default)

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki generate-key-pair

```
user@host> request security pki generate-key-pair type [xxx] size [xxx] certificate-id
test
Generated key pair test, key size [xxx] bits
```

Release Information

Command introduced in Junos OS Release 11.1.

Options to support Elliptic Curve Digital Signature Algorithm (ECDSA) added in Junos OS Release 12.1X45-D10.

521 option to support ECDSA introduced in Junos OS Release 19.1R1 on SRX5000 line with SRX5K-SPC3 card.

RELATED DOCUMENTATION

PKI Components In Junos OS | 33

request security pki key-pair export

IN THIS SECTION

Syntax

```
request security pki key-pair export certificate-id certificate-id filename filename
<passphrase string>
< type (der | pem)>
```

Description

Export the keypair for an end-entity (EE) certificate. The exported keypair is encrypted and can be imported along with the EE certificate. Using the CLI `request security pki key-pair export` command, you can export the pki key-pairs file as a backup or to check the file for troubleshooting purposes. We recommend denying access to the CLI `request security pki key-pair export` command to all users and restrict this command only to the privileged users.

Options

certificate-id *certificate-id*	Name of the local digital certificate.

filename *filename*	Target directory location and filename of the CA digital certificate.
passphrase *passphrase*	(Optional) Passphrase to protect the keypair data for PEM format. The passphrase can be up to 64 characters. If specified, the passphrase must be used when importing the keypair.
type (der | pem)	(Optional) Type of format, either DER or PEM. PEM is the default.

Required Privilege Level

maintenance

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 15.1X49-D60.

RELATED DOCUMENTATION

request security pki local-certificate export | 1752

request security pki local-certificate enroll cmpv2

IN THIS SECTION

Syntax

```
request security pki local-certificate enroll cmpv2
    ca-dn  subject-dn
    ca-profile  ca-profile name
    ca-reference  reference
    ca-secret  shared-secret
    certificate-id certificate-id-name
    domain-name domain-name
    email email-address
     ip-address ip-address
    ipv6-address ipv6-address
    subject subject-distinguished-name
```

Description

Enroll and install a local digital certificate online by using CMPv2. This command loads both end-entity (EE) and CA certificates based on the CA server configuration. Certificate revocation list (CRL) or Online Certificate Status Protocol (OCSP) can be used to check the revocation status of a certificate.

Options

ca-dn *subject-dn*

The distinguished name (DN) of the CA enrolling the EE certificate must be specified during enrollment. This optional parameter is mandatory if the CA certificate is not already enrolled. If the CA certificate is already enrolled, the subject DN is extracted from the CA certificate.

ca-profile *ca-profile-name*

CA profile name.

ca-reference *reference*

Out-of-band reference value received from the CA server.

ca-secret *shared-secret*

Out-of-band secret value received from the CA server.

certificate-id *certificate-id-name*

Name of the local digital certificate and the public/private key pair.

domain-name *domain-name*

Fully qualified domain name (FQDN). The FQDN provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.

email *email-address*

E-mail address of the certificate holder.

ip-address *ip-address*

IP address of the router.

ipv6-address *ipv6-address*

IPv6 address of the router for the alternate subject.

subject *subject-distinguished-name*

Distinguished Name (DN) format that contains the domain component, common name, department, serial number, company name, state, and country in the following format: DC, CN, OU, O, SN, L, ST, C.

- `DC`—Domain component
- `CN`—Common name
- `OU`—Organizational unit name
- `O`—Organization name
- `SN`—Serial number of the device

  If you define SN in the subject field without the serial number, then the serial number is read directly from the device and added to the certificate signing request (CSR).

- `ST`—State

- `C`—Country

Required Privilege Level

maintenance and security

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

command-name

```
user@host> request security pki local-certificate enroll cmpv2 ca-profile root-552 ca-dn
DC=example,CN=root-552 certificate-id tc552 email tc552-root@example.net domain-name example.net
ip-address 192.0.2.22 ca-secret example ca-reference 51892 subject CN=example,OU=SBU,O=552-22

Certificate enrollment has started. To view the status of your enrollment, check the public key
infrastructure log (pkid) log file at /var/log/pkid.
```

Release Information

Command introduced in Junos OS Release 15.1X49-D40.

RELATED DOCUMENTATION

show security pki local-certificate (View) | 1972

clear security pki local-certificate (Device) | 1708

request security pki local-certificate enroll scep

IN THIS SECTION

Syntax

```
request security pki local-certificate enroll scep
    ca-profile  ca-profile name
    certificate-id certificate-id-name
    challenge-password challenge-password
    digest (sha-1 | sha-256)
    domain-name domain-name
    email email-address
    ip-address ip-address
    ipv6-address ipv6-address
    logical-system (logical-system-name | all)
    scep-digest-algorithm (md5 | sha-1)
    scep-encryption-algorithm (des | des3)
    subject subject-distinguished-name
```

Release Information

Command introduced in Junos OS Release 9.1. Serial number (SN) option added to the subject string output field in Junos OS Release 12.1X45. `scep` keyword and `ipv6-address` option added in Junos OS Release 15.1X49-D40.

Starting in Junos OS Release 20.1R1 on vSRX Virtual Firewall 3.0, you can safeguard the private keys used by PKID and IKED using Microsoft Azure Key Vault hardware security module (HSM) service. You can establish a PKI based VPN tunnel using the keypairs generated at the HSM. The hub `certificate-id` option under certificate-id is not available for configuration after generating HSM key-pair.

Starting in Junos OS Release 20.4R1 on vSRX Virtual Firewall 3.0, you can safeguard the private keys used by PKID and IKED using AWS Key Management Service (KMS). You can establish a PKI based VPN tunnel using the keypairs generated by the KMS. The hub `certificate-id` option under certificate-id is not available for configuration after generating PKI key-pair.

Starting in Junos OS Release 22.4R2, `logical-system` is introduced in the statement for PKI SCEP certificate enrollment.

Description

Enroll and install a local digital certificate online by using Simple Certificate Enrollment Protocol (SCEP).

If you enter the `request security pki local-certificate enroll` command without specifying the `scep` or `cmpv2` keyword, SCEP is the default method for enrolling a local certificate.

Options

ca-profile *ca-profile-name* CA profile name.

certificate-id *certificate-id-name* Name of the local digital certificate and the public/private key pair.

challenge-password *password* Password set by the administrator and normally obtained from the SCEP enrollment webpage of the CA. The password is maximum 256 characters in length. You can enforce the limit to the required characters.

digest (sha-1 | sha-256) Hash algorithm used for signing RSA certificates, either SHA-1 or SHA-256. SHA-1 is the default.

domain-name *domain-name*

Fully qualified domain name (FQDN). The FQDN provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.

email *email-address*

E-mail address of the certificate holder.

ip-address *ip-address*

IP address of the router.

ipv6-address *ipv6-address*

IPv6 address of the router for the alternate subject.

logical-system (*logical-system-name* | all)

Name of the logical system or all. This is optional.

scep-digest-algorithm (md5 | sha-1)

Hash algorithm digest, either MD5 or SHA-1; SHA-1 is the default.

scep-encryption-algorithm (des | des3)

Encryption algorithm, either DES or DES3; DES3 is the default.

subject *subject-distinguished-name*

Distinguished Name (DN) format that contains the domain component, common name, department, serial number, company name, state, and country in the following format: DC, CN, OU, O, SN, L, ST, C.

- `DC`—Domain component
- `CN`—Common name
- `OU`—Organizational unit name
- `O`—Organization name
- `SN`—Serial number of the device

  If you define SN in the subject field without the serial number, then the serial number is read directly from the device and added to the certificate signing request (CSR).
- `ST`—State
- `C`—Country

Required Privilege Level

maintenance and security

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

command-name

```
user@host> request security pki local-certificate enroll scep certificate-id r3-entrust-scep ca-
profile entrust domain-name router3.example.net subject
"CN=router3,OU=Engineering,O=example,C=US" challenge-password 123
```

```

Certificate enrollment has started. To view the status of your enrollment, check the public key
infrastructure log (pkid) log file at /var/log/pkid. Please save the challenge-password for
revoking this certificate in future.  Note that this password is not stored on the router.
```

Sample Output

Sample output for vSRX Virtual Firewall 3.0

```
user@host> request security pki generate-key-pair certificate-id example
```

```
Generated key pair example, key size 2048 bits
```

```
user@host> request security pki local-certificate enroll certificate-id ?
```

```
Possible completions:
<certificate-id> Certificate identifier
example
```

```
user@host> request security pki generate-key-pair certificate-id Hub
```

```
error: Failed to generate key pair at HSM. Found a key with the same name at HSM. Use a
different certificate id next time. Refer to PKID logs for more details
```

RELATED DOCUMENTATION

- request security pki local-certificate enroll cmpv2
- show security pki local-certificate (View)
- clear security pki local-certificate (Device)

request security pki local-certificate export

IN THIS SECTION

Syntax

```
request security pki local-certificate export
```

Description

Export a generated self-signed certificate from the default location (var/db/certs/common/local) to a specific location within the device.

Options

certificate id *certificate-id-name*	Name of the local digital certificate.
filename *path/filename*	Target directory location and filename of the CA digital certificate.
type (der | pem)	Certificate format: DER (distinguished encoding rules) or PEM (privacy-enhanced mail).

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki local-certificate export

```
user@host> request security pki local-certificate export filename /var/tmp/my-cert.pem
certificate-id nss-cert type pem
certificate exported successfully
```

Release Information

Command introduced in Junos OS Release 12.1.

RELATED DOCUMENTATION

PKI Components In Junos OS | 33

request security pki local-certificate generate-self-signed (Security)

IN THIS SECTION

Syntax

```
request security pki local-certificate generate-self-signed certificate-id certificate-id-
namedomain-name domain-name subject subject-distinguished-name
<add-ca-constraint>
<digest (sha1 | sha256)>
<email email-address>
<ip-address ipv4-address>
<ipv6-address ipv6-address>
```

Description

Manually generate a self-signed certificate for the given distinguished name.

Options

`certificate-id` *`certificate-id-name`*—Name of the certificate and the public/private key pair.

`domain-name` *`domain-name`*—Fully qualified domain name (FQDN) provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.

`subject` *`subject-distinguished-name`*—Distinguished name format contains the following information:

- `DC`—Domain component
- `CN`—Common name
- `OU`—Organizational unit name
- `O`—Organization name
- `L`—Locality
- `ST`—State
- `C`—Country

`add-ca-constraint`—(Optional) Specifies that the certificate can be used to sign other certificates.

`digest`—(Optional) Hash algorithm used to sign the certificate.

- `sha1`—SHA-1 digest (default)
- `sha256`—SHA-256 digest

Starting in Junos OS Release 18.1R3, the default encryption algorithm that is used for validating automatically and manually generated self-signed PKI certificates is Secure Hash Algorithm 256 (SHA-256). Prior to Junos OS Release 18.1R3, SHA-1 is used as default encryption algorithm.

`email` *`email-address`*—(Optional) E-mail address of the certificate holder.

`ip-address` *`ipv4-address`*`–(Optional) Static IPv4 address of the device.`

`ipv6-address` *`ipv6-address`*`–(Optional) Static IPv6 address of the device.`

Required Privilege Level

maintenance and security

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki local-certificate generate-self-signed certificate-id self-cert subject cn=abc domain-name example.net email mholmes@example.net

```
user@host> request security pki local-certificate generate-self-signed certificate-id self-cert
subject cn=abc domain-name example.net email mholmes@example.net
Self-signed certificate generated and loaded successfully
```

Release Information

Command introduced in Junos OS Release 9.1.

Support for `digest` option added in Junos OS Release 12.1X45-D10.

Support for `ipv6-address` option added in Junos OS Release 22.1R1.

RELATED DOCUMENTATION

clear security pki local-certificate (Device) | 1708

show security pki local-certificate (View) | 1972

request security pki local-certificate load

IN THIS SECTION

Syntax

```
request security pki local-certificate load filename ssl_proxy_ca.crt key ssl_proxy_ca.key
certificate-id certificate id
```

Description

Manually load a local digital certificate from a specified location.

Options

filename Filename that contains the certificate to load

key File pathname that contains the private key/key-pair to loaded

certificate-id Name of the certificate identifier

Starting in Junos OS Release 19.1R1, a commit check is added to prevent user from adding `.`, `/`, `%`, and space in a certificate identifier while generating a local or remote certificates or a key pair.

Required Privilege Level

maintenance and security

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki local-certificate load

```
user@host> request security pki local-certificate load filename cert_name.crt key key_name.key
certificate-id test
Local certificate cert_name.crt loaded successfully
```

Release Information

Command introduced in Junos OS Release 11.4.

RELATED DOCUMENTATION

show security pki local-certificate (View) | **1972**

clear security pki local-certificate (Device) | **1708**

request security pki local-certificate verify (Security) | **1763**

PKI Components In Junos OS | **33**

request security pki local-certificate re-enroll cmpv2

IN THIS SECTION

Syntax

```
request security pki local-certificate re-enroll cmpv2 certificate-id certificate-id
<ca-profile-name ca-profile>
<re-generate-keypair>
```

Description

Manually reenroll an end-entity (EE) certificate with Certificate Management Protocol version 2 (CMPv2). This command allows the administrator to initiate renewal of the EE certificate using CMPv2 and can be used in conjunction with the `set security pki auto-re-enrollment cmpv2` automatic enrollment configuration.

Options

certificate-id ***certificate-id-name*** Name of the local digital certificate.

ca-profile-name ***ca-profile-name*** (Optional) CA profile name.

re-generate-keypair	(Optional) Generate a PKI public/private key pair for the EE certificate. Key generation might take a few seconds.

Required Privilege Level

maintenance and security

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 15.1X49-D60.

RELATED DOCUMENTATION

request security pki local-certificate enroll cmpv2 | **1743**

request security pki local-certificate re-enroll scep

IN THIS SECTION

Syntax

```
request security pki local-certificate re-enroll scep certificate-id certificate-id
<ca-profile-name ca-profile>
<challenge-password password>
<re-generate-keypair>
<scep-digest-algorithm (md5 | sha-1)>
<scep-encryption-algorithm (des | des3)>
```

Description

Manually reenroll an end-entity (EE) certificate with Simple Certificate Enrollment Protocol (SCEP). This command allows the administrator to initiate renewal of the EE certificate using SCEP and can be used in conjunction with the `set security pki auto-re-enrollment scep` automatic enrollment configuration.

Starting in Junos OS Release 20.1R1 on vSRX Virtual Firewall 3.0, you can safeguard the private keys used by PKID and IKED to establish a PKI based VPN tunnel using the keypairs generated at the Microsoft Azure Key Vault hardware security module (HSM) service and starting in Junos OS Release 20.4R1 on vSRX Virtual Firewall 3.0, the same feature is supported through AWS Key Management Service (KMS).

You cannot manually re-enroll the local certificates with the “re-generate key-pair” option. An error message is displayed.

Warning message upon re-enrollment - sample output:

```
[edit]
root@vsrx-1# ...te-id hsm1 ca-profile azure-ca challenge-password juniper re-generate-keypair
error: HSM Error: Re-enrollment is not allowed with re-generate key-pair option.
```

Options

certificate-id *certificate-id-name*

Name of the local digital certificate.

ca-profile-name *ca-profile-name*

(Optional) CA profile name.

challenge-password *password*

Password set by the administrator and normally obtained from the SCEP enrollment webpage of the CA. The password is 16 characters in length.

re-generate-keypair

(Optional) Generate a PKI public/private key pair for the EE certificate.

Key generation might take a few seconds.

scep-digest-algorithm

(Optional) Hash algorithm digest, either MD5 or SHA-1; SHA-1 is the default.

scep-encryption-algorithm

(Optional) Encryption algorithm, either DES or DES3; DES3 is the default.

Required Privilege Level

maintenance and security

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 15.1X49-D60.

RELATED DOCUMENTATION

request security pki local-certificate enroll scep

request security pki local-certificate verify (Security)

IN THIS SECTION

Syntax

```
request security pki local-certificate verify certificate-id  certificate-id-name
```

Description

Verify the validity of the local digital certificate identifier.

Options

`certificate-id` *`certificate-id-name`* — Name of the local digital certificate identifier.

Required Privilege Level

maintenance and security

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki local-certificate verify certificate-id bme1 (not downloaded)

You receive the following response before the certificate revocation list (CRL) is downloaded:

```
user@host> request security pki local-certificate verify certificate-id bme1
Local certificate bme1: CRL verification in progress. Please check the PKId debug logs for
completion status
```

request security pki local-certificate verify certificate bme1 (downloaded)

You receive the following response after the certificate revocation list (CRL) is downloaded:

```
user@host> request security pki local-certificate verify certificate-id bme1
Local certificate bme1 verification success
```

request security pki local-certificate verify certificate-id pc_hub (Verify certificate revoke status on MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

You receive the following response after the local certificate is revoked:

```
user@host> request security pki local-certificate verify certificate-id pc_hub
Local cert pc_hub verification failed. local cert is revoked
```

request security pki local-certificate verify certificate-id hub_cert1 (Verify local certificate status when an intermediate CA is deleted for SRX Series Firewalls and vSRX Virtual Firewall)

You receive the following response when an intermediate CA certificate is deleted:

```
user@host> request security pki local-certificate verify certificate-id hub_cert1
Local cert hub_cert1 verification failed. Cannot build cert chain.
```

request security pki local-certificate verify certificate-id pc1 (Verify enrolled local certificate present in MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

You receive the following response when the local certificate is missing:

```
user@host> request security pki local-certificate verify certificate-id pc1
Local cert pc1 verification failed. local cert is missing
```

request security pki local-certificate verify certificate-id localcert-root (Verify local certificate status when the CA is unreachable for MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

You receive the following response when a CA is not reachable or CRL download has failed.

```
user@host> request security pki local-certificate verify certificate-id localcert-root
Local Cert localcert-root Verification Failed. Unreachable CA or CRL Download Failed
```

Release Information

Command introduced in Junos OS Release 8.5.

RELATED DOCUMENTATION

request security pki local-certificate load | 1756

request security pki ca-profile-group-status (MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall) | 1735

request security pki verify-integrity-status

IN THIS SECTION

Syntax

```
request security pki verify-integrity-status
```

Description

Verify the integrity of public key infrastructure (PKI) files. This feature is supported only on SRX5400, SRX5600, and SRX5800 devices and vSRX Virtual Firewall instances.

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security pki verify-integrity-status

```
user@host> request security pki verify-integrity-status
All PKI objects: verification success
```

Release Information

Command introduced in Junos OS Release 11.2.

Do not use this command for non-FIPS or Common Criteria releases. We recommend that you do not use this command for any Junos OS Release 15.1X49-D40 or later releases.

request security re-distribution ipsec-vpn

IN THIS SECTION

Syntax

```
request security re-distribution ipsec-vpn
gateway-name <gateway-name>
fpc <fpc-number>
pic <pic-number>
[thread-id <tid>]
[remote-id <rid>]
```

Description

Redistribute the tunnels that belongs to a Auto VPN or site-to-site gateway to a new processing unit.

This command migrates the tunnels only once and is valid only for 30 minutes, if the peer does not bring up the tunnel(s) immediately. After execution of the command, subsequent tunnels for the peer is established on the same FPC, PIC, and thread-id (only if specified).

In case of Auto VPN gateways, once the tunnels are brought down, it is expected that peer re-establishes the tunnel.

This command causes traffic disruption when used on an already established tunnel. If the command is used on a tunnel which is already anchored on the destination processing unit, it will not tear down the tunnel and re-establish it.

This feature is supported only on SRX5K-SPC3 (SPC3) card and in mixed-mode (SPC3 or SRX5K-SPC-4-15-320 (SPC2) cards).

When a tunnel goes down, you can use only the syslog to trace why a tunnel is anchored on a different processing unit.

If you want to migrate the tunnel back to the previous FPC or PIC (that is, default profile), you can either redistribute the tunnel again or run the `clear security ike security-associations index` *`SA-index-number`* command.

Options

gateway-name *gateway-name* Name of the gateway.

fpc *fpc-number* FPC slot number (0..63).

pic *pic-number* PIC slot number (0..3).

thread-id *tid* (Optional) Thread ID number. Only valid for SPC3. (1..27)

remote-id *rid* If you provide Auto VPN as a gateway, then it is mandatory to provide the *remote-id*. If you provide site-to-site as a gateway, then you need not provide the *remote-id*.

Required Privilege Level

maintenance

Release Information

Command introduced in Junos OS Release 20.4R1.

RELATED DOCUMENTATION

show security ipsec tunnel-distribution | **1949**

show security re-distribution ipsec-vpn | **1980**

request security pki sync-from-peer

IN THIS SECTION

Syntax

```
request security pki sync-from-peer
```

Description

Synchronize the PKI file system on the peer node in a Multinode High Availability setup. You can use this command to replicate the PKI directory in the remote node to your local node. Replicating PKI directory is helpful when one of the two nodes or ICL goes down.

Note that you can run this command only you've enabled Multinode High Availabiliy.

Consider a set up with node 0 (local node) and node 1 (remote node). To replicate the PKI directory of the remote node (node 1), run this command in your local node (node 0).
When you run this command on your local node, all the local PKI files are deleted and replaced by the remote node PKI directory. Hence, be sure on which node you are executing this command. After running this command, we recommend you to verify whether the files are synchronized between the two nodes.

Required Privilege Level

maintenance

Output Fields

This command produces no output.

Sample Output

request security pki sync-from-peer

```
user@host> request security pki sync-from-peer
File syncing is in progress... This will take a few seconds. Please confirm that the files are
synched. If not, run this command once again.
```

Release Information

Command introduced in Junos OS Release 20.4R1.

RELATED DOCUMENTATION

- ike (High Availability) | **1521**
- ipsec (High Availability) | **1544**
- High-Availability (Chassis)
- Multinode High Availability

show network-access address-assignment pool (View)

IN THIS SECTION

Syntax

```
show network-access address-assignment pool name
```

Description

Display information summary about a specific pool.

Required Privilege Level

view

Output Fields

Table 126 on page 1773 lists the output fields for the `show network-access address-assignment pool` command. Output fields are listed in the approximate order in which they appear.

Table 126: show network-access address-assignment pool Output Fields

Field Name	Field Description
`IP address`	IP address assigned to a client.
`Hardware address`	MAC address of the client. For XAuth clients, the value is NA.
`Host/User`	For static IP address assignment, the user name and profile are displayed in the format *username@profile*. If the client is assigned an IP address from an address pool and a user name exists, the user name is displayed. For DHCP applications, if the host name is configured the host name is displayed; otherwise NA is displayed.
`Type`	Either XAuth or DHCP attributes are configured.

Sample Output

command-name

```
user@host> show network-access address-assignment pool xauth1
IP address       Hardware address        Host/User                 Type
192.0.2.1         NA                      jason@dvpn-auth           XAUTH
192.0.2.2         NA                      jacky                     XAUTH
192.0.2.3         00:00:5E:00:53:01       host1                     DHCP
192.0.2.4         00:00:5E:00:53:02       NA                        DHCP
```

Release Information

Command introduced in Junos OS Release 10.4.

show security dynamic-policies

IN THIS SECTION

Syntax

```
show security dynamic-policies [detail] [from-zone zone] [scope-id id] [to-zone zone]
```

Description

Display dynamic policies downloaded on the group member. This command is supported on SRX100, SRX110, SRX210, SRX220, SRX240, and SRX650 devices.

Options

- none—Display basic information about all policies installed on the group member.
- `detail`—(Optional) Display a detailed view of all of the policies installed on the group member.
- `from-zone`—(Optional) Display information about the policies installed on the group member for the specified source zone.
- `scope-id`—(Optional) Display information about the policies installed on the group member for the specified policy identifier.
- `to-zone`—(Optional) Display information about the policies installed on the group member for the specified destination zone.

Required Privilege Level

view

Output Fields

Table 127 on page 1775 lists the output fields for the `show security dynamic-policies` command. Output fields are listed in the approximate order in which they appear.

Table 127: show security dynamic-policies Output Fields

Field Name	Field Description
`Policy`	Name of the applicable Policy.

Table 127: show security dynamic-policies Output Fields *(Continued)*

Field Name	Field Description
`State`	Status of the policy: • `enabled`: The policy can be used in the policy lookup process, which determines access rights for a packet and the action taken in regard to it. • `disabled`: The policy cannot be used in the policy lookup process, and therefore it is not available for access control.
`Index`	An internal number associated with the policy.
`Scope Policy`	Policy identifier.
`Sequence number`	Number of the policy within a given context. For example, three policies that are applicable in a from-zoneA-to-zoneB context might be ordered with sequence numbers 1, 2, and 3. Also, in a from-zoneC-to-zoneD context, four policies might have sequence numbers 1, 2, 3, and 4.
`Source addresses`	For standard display mode, the names of the source addresses for a policy. Address sets are resolved to their individual names. (In this case, only the names are given, not their IP addresses.) For detail display mode, the names and corresponding IP addresses of the source addresses for a policy. Address sets are resolved to their individual address name-IP address pairs.
`Destination addresses`	Name of the destination address (or address set) as it was entered in the destination zone's address book. A packet's destination address must match this value for the policy to apply to it.

Table 127: show security dynamic-policies Output Fields *(Continued)*

Field Name	Field Description
`Application`	Name of a preconfigured or custom application whose type the packet matches, as specified at configuration time. • `IP protocol`: The IP protocol used by the application—for example, TCP, UDP, ICMP. • `ALG`: If an ALG is associated with the session, the name of the ALG. Otherwise, 0. • `Inactivity timeout`: Elapse time without activity after which the application is terminated. • `Source port range`: The low-high source port range for the session application. • `Destination port range`: The low-high destination port range for the session application.
`action-type`	Must be permit.
`Policy Type`	Must be dynamic.
`From zone`	Name of the source zone.
`To zone`	Name of the destination zone.
`Tunnel`	Tunnel name, type (IPsec), and index number.

Sample Output

show security dynamic-policies

```
user@host> show security dynamic-policies
Policy: policy_forward-0001, State: enabled, Index: 1048580, Scope Policy: 4
```

```
  Sequence number: 1
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
    Applications: Unknown
action-type: permit, tunnel:
Policy: policy_forward-0002, State: enabled, Index: 2097156, Scope Policy: 4
  Sequence number: 2
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
    Applications: Unknown
action-type: permit, tunnel:
```

Sample Output

show security dynamic-policies detail

```
user@host> show security dynamic-policies detail
Policy: policy_forward-0001, action-type: permit, State: enabled, Index: 1048580,AI: disabled,
Scope Policy: 4
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Host, To zone: untrust
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [23-23]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1001
Policy: policy_backward-0001, action-type: permit, State: enabled, Index: 1048582,AI: disabled,
Scope Policy: 6
  Policy Type: Dynamic
  Sequence number: 1
  From zone: untrust, To zone: Host
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1003
Policy: policy_internal-0001, action-type: permit, State: enabled, Index: 1048583,AI: disabled,
Scope Policy: 7
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Internal, To zone: Host
  Source addresses:192.168.1.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1005
Policy: policy_external-0001, action-type: permit, State: enabled, Index: 1048584,AI: disabled,
Scope Policy: 8
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Internal, To zone: untrust
  Source addresses:192.168.1.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1006
Policy: policy_forward-0002, action-type: permit, State: enabled, Index: 2097156,AI: disabled,
Scope Policy: 4
  Policy Type: Dynamic
  Sequence number: 2
  From zone: Host, To zone: untrust
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1002
Policy: policy_backward-0002, action-type: permit, State: enabled, Index: 2097158,AI: disabled,
Scope Policy: 6
  Policy Type: Dynamic
  Sequence number: 2
  From zone: untrust, To zone: Host
  Source addresses:192.168.10.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [23-23]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1004
```

Sample Output

show security dynamic-policies from-zone Internal

```
user@host> show security dynamic-policies  from-zone Internal
Policy: policy_internal-0001, State: enabled, Index: 1048583, Scope Policy: 7
  Sequence number: 1
    Applications: Unknown
action-type: permit, tunnel:
Policy: policy_external-0001, State: enabled, Index: 1048584, Scope Policy: 8
  Sequence number: 1
    Applications: Unknown
action-type: permit, tunnel:
```

Sample Output

show security dynamic-policies scope-id 8 from-zone Internal

```
user@host> show security dynamic-policies scope-id 8 from-zone Internal
Policy: policy_external-0001, State: enabled, Index: 1048584, Scope Policy: 8
  Sequence number: 1
    Applications: Unknown
action-type: permit, tunnel:
```

Sample Output

show security dynamic-policies detail from-zone Internal

```
user@host> show security dynamic-policies detail from-zone Internal
Policy: policy_internal-0001, action-type: permit, State: enabled, Index: 1048583,AI: disabled,
Scope Policy: 7
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Internal, To zone: Host
  Source addresses:192.168.1.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1005
Policy: policy_external-0001, action-type: permit, State: enabled, Index: 1048584,AI: disabled,
Scope Policy: 8
  Policy Type: Dynamic
  Sequence number: 1
  From zone: Internal, To zone: untrust
  Source addresses:192.168.1.0/24
  Destination addresses:192.168.20.0/24
  Application: Unknown
    IP protocol: 6, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [80-80]
  Tunnel: Test Tunnel, Type: IPSec, Index: 1006
```

Sample Output

show security dynamic-policies detail from-zone Internal to-zone Host

```
user@host> show security dynamic-policies detail from-zone Internal to-zone Host
Policy: policy_internal-0001, action-type: permit, State: enabled, Index: 1048583,AI: disabled,
Scope Policy: 7
  Policy Type: Dynamic
```

```
Sequence number: 1
From zone: Internal, To zone: Host
Source addresses:192.168.1.0/24
Destination addresses:192.168.20.0/24
Application: Unknown
  IP protocol: 6, ALG: 0, Inactivity timeout: 0
    Source port range: [0-0]
    Destination port range: [80-80]
Tunnel: Test Tunnel, Type: IPSec, Index: 1005
```

Release Information

Command introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

show security policies

Group VPNv2 Overview | 758

show security group-vpn member ike security-associations

IN THIS SECTION

Syntax

```
show security group-vpn member ike security-associations [brief | detail] [index sa-index] [peer-
ipaddress]
```

Description

Display IKE security associations (SAs) for group members. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

- none—Display summary information about all IKE SAs for the group members.
- `brief`—(Optional) Display summary output.
- `detail`—(Optional) Display detailed output.
- `index`*sa-index*—(Optional) Display detailed information about the specified SA identified by index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.
- *peer-ipaddress*—(Optional) Display information about the SA with the specified peer.

Required Privilege Level

view

Output Fields

Table 128 on page 1784 lists the output fields for the `show security group-vpn member ike security-associations` command. Output fields are listed in the approximate order in which they appear.

Table 128: show security group-vpn member ike security-associations Output Fields

Field Name	Field Description
`Index`	Index number of an SA. This number is an internally generated number you can use to display information about a single SA.
`State`	State of the IKE security associations: • `DOWN`—SA has not been negotiated with the peer. • `UP`—SA has been negotiated with the peer.
`Initiator cookie`	Random number, called a cookie, which is sent to the remote node when the IKE negotiation is triggered.
`Responder cookie`	Random number generated by the remote node and sent back to the initiator as a verification that the packets were received. A cookie is aimed at protecting the computing resources from attack without spending excessive CPU resources to determine the cookie's authenticity.
`Mode`	Negotiation method agreed on by the two IPsec endpoints, or peers, used to exchange information between themselves. Each exchange type determines the number of messages and the payload types that are contained in each message. The modes, or exchange types, are • `main`—The exchange is done with six messages. This mode or exchange type encrypts the payload, protecting the identity of the neighbor. The authentication method used is displayed: preshared keys or certificate. • `aggressive`—The exchange is done with three messages. This mode or exchange type does not encrypt the payload, leaving the identity of the neighbor unprotected.
`Remote Address`	IP address of the destination peer with which the local peer communicates.

Table 128: show security group-vpn member ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`IKE Peer`	IP address of the destination peer with which the local peer communicates.
`Exchange type`	Negotiation method agreed on by the two IPsec endpoints, or peers, used to exchange information between themselves. Each exchange type determines the number of messages and the payload types that are contained in each message. The modes, or exchange types, are • `main`—The exchange is done with six messages. This mode or exchange type encrypts the payload, protecting the identity of the neighbor. The authentication method used is displayed: preshared keys or certificate. • `aggressive`—The exchange is done with three messages. This mode or exchange type does not encrypt the payload, leaving the identity of the neighbor unprotected.
`Authentication method`	Method the server uses to authenticate the source of IKE messages: • `pre-shared-keys`—Preshared key for encryption and decryption that both participants must have before beginning tunnel negotiations.
`Local`	Address of the local peer.
`Lifetime`	Number of seconds remaining until the IKE SA expires.

Table 128: show security group-vpn member ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`Algorithms`	Internet Key Exchange (IKE) algorithms used to encrypt and secure exchanges between the peers during the IPsec Phase 2 process: • `Authentication`—Type of authentication algorithm used. • `sha-256`—Secure Hash Algorithm 256 authentication. • `sha-384`—Secure Hash Algorithm 384 authentication. • `Encryption`—Type of encryption algorithm used. • `aes-256-cbc`—Advanced Encryption Standard (AES) 256-bit encryption. • `aes-192-cbc`— AES192-bit encryption • `aes-128-cbc`—AES 128-bit encryption.
`Traffic statistics`	• `Input bytes`–Number of bytes received. • `Output bytes`–Number of bytes transmitted. • `Input packets`–Number of packets received. • `Output packets`–Number of packets transmitted.

Sample Output

show security group-vpn member ike security-associations

```
user@host> show security group-vpn member ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
4736345 UP     70611c65603d53da  6e0888777ad10f8d  Main           192.0.2.3
```

Sample Output

show security group-vpn member ike security-associations detail

```
user@host> show security group-vpn member ike security-associations detail
IKE peer 192.0.2.5, Index 5824842, Gateway Name: group1_2
  Role: Initiator, State: UP
  Initiator cookie: fc866556b8afe4cd, Responder cookie: 1238de6b8a89de44
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 192.0.2.7:848, Remote: 192.0.2.5:848
  Lifetime: Expires in 2 seconds
  Peer ike-id: 192.0.2.5
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :                 2044
   Output bytes  :                  900
   Input  packets:                    7
   Output packets:                    7
  Flags: IKE SA is created
```

Release Information

Command introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

clear security group-vpn member ike security-associations | **1681**

Group VPNv2 Overview

show security group-vpn member ipsec inactive-tunnels

IN THIS SECTION

Syntax

```
show security group-vpn member ipsec inactive-tunnels <brief> <detail> <group-id group-id>
```

Description

Show inactive Group VPNs. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

none	Display information for all groups.
brief	(Optional) Display summary output.

detail	(Optional) Display detailed output.
group-id ***group-id***	(Optional) Display information for the specified group identifier.

Required Privilege Level

view

Output Fields

Table 129 on page 1789 lists the output fields for the `show security group-vpn member ipsec inactive-tunnels` command. Output fields are listed in the approximate order in which they appear.

Table 129: show security group-vpn member ipsec inactive-tunnels Output Fields

Field Name	Field Description
Server	Server on which group member is registered.
Port	UDP port number.
GId	Group identifier.
lsys	Logical system.

Table 129: show security group-vpn member ipsec inactive-tunnels Output Fields *(Continued)*

Field Name	Field Description
Reason	Reason that the tunnel is inactive: • The tunnel was cleared through the CLI. • The hard lifetime has expired. • There are too many TEKs. • There was a configuration change. • There was an SA installation error. • The TEK is stale. • The tunnel was deleted from the server.
Virtual-system	Logical system name.
Group VPN Name	Name of the Group VPN.
Local Gateway	IP address of the local IKE gateway.
GDOI Server	IP address of the group server.
Group Id	Group identifier.
Recovery Probe	Status of the recovery probe, either enabled or disabled (default).
DF-bit	Fragmentation of IPsec traffic on the group member—clear (default), copy, or set.
Stats	Statistics for GDOI groupkey-pull and groupkey-push exchanges, server failovers, deletes received, number of times the maximum number of keys and policies were exceeded, and the number of unsupported algorithms received.

Table 129: show security group-vpn member ipsec inactive-tunnels Output Fields *(Continued)*

Field Name	Field Description
Down Reason	Reason that the tunnel is inactive: • The tunnel was cleared through the CLI. • The hard lifetime has expired. • There are too many TEKs. • There was a configuration change. • There was an SA installation error. • The TEK is stale. • The tunnel was deleted from the server. • The tunnel is not initiated.

Sample Output

show security group-vpn member ipsec inactive-tunnels

```
user@host> show security group-vpn member ipsec inactive-tunnels
  Total inactive tunnels: 1
  Server           Port  GId lsys  Reason
  192.168.1.50       848   1000 root uninitiated
```

show security group-vpn member ipsec inactive-tunnels detail

```
user@host> show security group-vpn member ipsec inactive-tunnels detail
  Virtual-system: root Group VPN Name: group1000
  Local Gateway: 192.168.1.101, GDOI Server: 192.168.1.50
  Group Id: 1000
  Recovery Probe: Disabled
  DF-bit: clear
```

```
    Stats:
        Pull Succeeded             :   0
        Pull Failed                :   8841
        Pull Timeout               :   7996
        Pull Aborted               :   0
        Push Succeeded             :   0
        Push Failed                :   0
        Server Failover            :   0
        Delete Received            :   0
        Exceed Maximum Keys(4)     :   0
        Exceed Maximum Policies(10):   0
        Unsupported Algo           :   0
    Down Reason: uninitiated
```

Release Information

Command introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

show security group-vpn member ipsec security-associations

IN THIS SECTION

Syntax

```
show security group-vpn member ipsec security-associations [brief | detail] [index sa-index]
```

Description

Display group VPN security associations (SAs) for a group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

- none—Display information about all group VPN SAs for the group member.
- `brief`—(Optional) Display summary output.
- `detail`—(Optional) Display detailed output.
- `index` *sa-index*—(Optional) Display detailed information about the specified SA identified by index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.

Required Privilege Level

view

Output Fields

Table 130 on page 1794 lists the output fields for the `show security group-vpn member ipsec security-associations` command. Output fields are listed in the approximate order in which they appear.

Table 130: show security group-vpn member ipsec security-associations

Field Name	Field Description
`Total active tunnels`	Total number of active IPsec tunnels.
`ID`	Index number of the SA. You can use this number to get additional information about the SA.
`Server`	IP address of the group server (remote gateway).
`Port`	If Network Address Translation-Traversal (NAT-T) is used, this value is 4500. Otherwise it is the standard IKE port, 500.
`Algorithm`	Cryptography used to secure exchanges between peers during the IKE Phase 2 negotiations includes • An authentication algorithm used to authenticate exchanges between the peers. Options are `sha-256` or `sha-384` • An encryption algorithm used to encrypt data traffic. Options are `aes-128`, `aes-192`, and `aes-256`.
`SPI`	Security parameter index (SPI) identifier. An SA is uniquely identified by an SPI.
`Life: sec/kb`	The lifetime of the SA, after which it expires, expressed either in seconds or kilobytes.
`GId`	Group identifier.
`vsys or Virtual-system`	The root system.

Table 130: show security group-vpn member ipsec security-associations *(Continued)*

Field Name	Field Description
`Local Gateway`	Gateway address of the local system.
`GDOI Server`	IP address of the group server.
`Local Identity`	Identity of the local peer so that its partner destination gateway can communicate with it. The value is specified as an IPv4 address, fully qualified domain name, e-mail address, or distinguished name.
`Remote Identity`	IPv4 address of the destination peer gateway.
`DF-bit`	State of the don't fragment bit: set or cleared.
Forward-policy-mismatch	Enable the support for forwarding policy-mismatched packets
`Policy name`	Name of the applicable policy.
`Direction`	Direction of the security association; it can be inbound or outbound.
`AUX-SPI`	Value of the auxiliary security parameter index. • When the value is AH or ESP, AUX-SPI is always 0. • When the value is AH+ESP, AUX-SPI is always a positive integer.
`Hard lifetime`	The hard lifetime specifies the lifetime of the SA. • `Expires in seconds`—Number of seconds left until the SA expires.
`Lifesize Remaining`	The lifesize remaining specifies the usage limits in kilobytes. If there is no lifesize specified, it shows unlimited. • `Expires in kilobytes`—Number of kilobytes left until the SA expires.

Table 130: show security group-vpn member ipsec security-associations *(Continued)*

Field Name	Field Description
Soft lifetime	The soft lifetime informs the IPsec key management system that the SA is about to expire. Each lifetime of a security association has two display options, hard and soft, one of which must be present for a dynamic security association. This allows the key management system to negotiate a new SA before the hard lifetime expires. • Expires in seconds—Number of seconds left until the SA expires.
Mode	Mode of the security association: • transport—Protects host-to-host connections. • tunnel—Protects connections between security gateways.
Protocol	Protocol supported. Transport mode supports Encapsulation Security Protocol (ESP).
Anti-replay service	State of the service that prevents packets from being replayed. It can be Enabled or Disabled.

Sample Output

show security group-vpn member ipsec security-associations

```
user@host> show security group-vpn member ipsec security-associations
  Total active tunnels: 2
  ID    Server            Port  Algorithm      SPI      Life:sec/kb  GId lsys
  <>49157 192.168.1.53     848   ESP:3des/sha1  c0792f86 114/  unlim   2000 root
  <>49156 192.168.1.53     848   ESP:aes-256/md5 7def169d 18/   unlim   2000 root
  <>49156 192.168.1.53     848   ESP:aes-256/md5 86c48448 146/  unlim   2000 root
```

Sample Output

show security group-vpn member ipsec security-associations detail

```
user@host> show security group-vpn member ipsec security-associations detail
  Virtual-system: root Group VPN Name: group2000
  Local Gateway: 192.168.1.70, GDOI Server: 192.168.1.53
  Group Id: 2000
  Routing Instance: vr1
  Recovery Probe: Enabled
  DF-bit: clear
Forward-policy-mismatch:Enabled

  Stats:
      Pull Succeeded            :   3
      Pull Failed               :   0
      Pull Timeout              :   6
      Pull Aborted              :   0
      Push Succeeded            :   1773
      Push Failed               :   0
      Server Failover           :   0
      Delete Received           :   0
      Exceed Maximum Keys(4)    :   0
      Exceed Maximum Policies(10):   0
      Unsupported Algo          :   0
  Flags:
      Rekey Needed:   no

    List of policies received from server:
    Tunnel-id: 49157
      Source IP: ipv4_subnet(any:900,[0..7]=192.168.1.0/24)
      Destination IP: ipv4_subnet(any:901,[0..7]=192.168.1.0/24)

      Direction: bi-directional, SPI: c0792f86
      Protocol: ESP, Authentication: sha1, Encryption: 3des
      Hard lifetime: Expires in 81 seconds, Activated
      Lifesize Remaining:  Unlimited
      Soft lifetime: Expired
      Mode: Tunnel, Type: Group VPN, State: installed
      Anti-replay service: D3P enabled, Window size: 3000 milliseconds
```

```
        Direction: bi-directional, SPI: a645b381
        Protocol: ESP, Authentication: sha1, Encryption: 3des
        Hard lifetime: Expires in 207 seconds, Activated in 51 seconds
        Lifesize Remaining:  Unlimited
        Soft lifetime: Expires in 117 seconds
        Mode: Tunnel, Type: Group VPN, State: installed
        Anti-replay service: D3P enabled, Window size: 3000 milliseconds
```

Release Information

Command introduced in Junos OS Release 10.2.

Command introduced in Junos OS Release 18.2R1 for MX-series.

RELATED DOCUMENTATION

clear security group-vpn member ipsec security-associations | **1683**

Group VPNv2 Overview | **758**

show security group-vpn member ipsec statistics

IN THIS SECTION

Syntax

```
show security group-vpn member ipsec statistics <index index>
```

Description

Show IPsec statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

none	Display information for all IPsec SAs.
index *index*	(Optional) Display detailed information about the specified SA, identified by index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.

Required Privilege Level

view

Output Fields

Table 131 on page 1800 lists the output fields for the `show security group-vpn member ipsec statistics` command. Output fields are listed in the approximate order in which they appear.

Table 131: show security group-vpn member ipsec statistics Output Fields

Field Name	Field Description
ESP Statistics	Numbers of encrypted and decrypted bytes and encrypted and decrypted packets.
AH Statistics	Numbers of input and output bytes and input and output packets.
Errors	Numbers of AH failures, replay errors, ESP authentication failures, ESP decryption failures, bad headers, and bad trailers.
D3P Statistics	Numbers of old timestamp packets, new timestamp packets, no timestamp packets, unexpected D3P header packets, invalid type packets, invalid length packets, and invalid next header packets.
Exclude Statistics	Numbers of created and invalidated sessions.
Dynamic Policy Statistics	Numbers of created and invalidated sessions.
Fail-Open Statistics	Numbers of created and invalidated sessions.
Fail-Close Statistics	Number of dropped packets.
Forward-policy-mismatch Statistics	Number of bypassed packets.

Sample Output

show security group-vpn member ipsec statistics

```
user@host> show security group-vpn member ipsec statistics
ESP Statistics:
  Encrypted bytes:            54712
  Decrypted bytes:            16800
  Encrypted packets:            381
  Decrypted packets:            200
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
D3P Statistics:
  Old timestamp packets:                  0
  New timestamp packets:                  0
  No timestamp packets:                   0
  Unexpected D3P header packets:          0
  Invalid type packets:                   0
  Invalid length packets:                 0
  Invalid next header packets:            0
Exclude Statistics:
  Created sessions:                       0
  Invalidated sessions:                   0
Dynamic Policy Statistics:
  Created sessions:                     381
  Invalidated sessions:                   0
Fail-Open Statistics:
  Created sessions:                       0
  Invalidated sessions:                   0
Fail-Close Statistics:
  Dropped packets:                        0
Forward-policy-mismatch Statistics:
  Input Packets:                  0
  Output packets:                 0
```

Release Information

Command introduced in Junos OS Release 15.1X49-D30.

Command introduced in Junos OS Release 18.2R1 for MX-series.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

show security group-vpn member kek security-associations

IN THIS SECTION

Syntax

```
show security group-vpn member kek security-associations [brief | detail | display xml] [index
sa-index] [peer-ipaddress]
```

Description

Display Group VPNv2 security associations (SAs) for a group member. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Group VPNv2 is the name of the Group VPN technology on MX5, MX10, MX40, MX80, MX240, MX480, and MX960 routers. Group VPNv2 is different from the Group VPN technology implemented on SRX Security Gateways.

For more information about Group VPN on SRX Security Gateway devices, see "Group VPNv2 Overview" on page 758.

Options

- none—Display information about all Group VPNv2 SAs for the group member.
- `brief`—(Optional) Display summary output.
- `detail`—(Optional) Display detailed output.
- `display xml`—(Optional) Display xml.
- `index` *sa-index*—(Optional) Display detailed information about the specified SA identified by index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.
- *peer-ipaddress*—(Optional) Display information about the SA with the specified peer.

Required Privilege Level

view

Output Fields

Table 132 on page 1804 lists the output fields for the `show security group-vpn member kek security-associations` command. Output fields are listed in the approximate order in which they appear.

Table 132: show security group-vpn member kek security-associations

Field Name	Field Description
`Index`	Index number of an SA. This number is an internally generated number you can use to display information about a single SA.
`Remote Address`	IP address of the destination peer with which the local peer communicates.
`State`	State of the KEK security associations: • `DOWN`—SA is not active. • `UP`—SA is active.
`Initiator cookie`	Random number, called a cookie, which is sent to the remote node when the IKE negotiation is triggered.
`Responder cookie`	Random number generated by the remote node and sent back to the initiator as a verification that the packets were received.
`SPI`	Security parameter index (SPI) identifier. An SA is uniquely identified by an SPI.
`GroupID`	Group identifier.
`KEK Peer`	IP address of the destination peer with which the local peer communicates.
`Role`	For the member, it is always responder.
`State`	State of the KEK security associations, which is always up.
`Authentication method`	RSA is the supported authentication method.
`Local`	Address of the local peer.

Table 132: show security group-vpn member kek security-associations *(Continued)*

Field Name	Field Description
`Remote`	Address of the remote peer.
`Lifetime`	Number of seconds remaining until the IKE SA expires.
`Algorithms`	Internet Key Exchange (IKE) algorithms used to encrypt and secure exchanges between the peers during the IPsec Phase 2 process: • `Sig-hash`—Type of authentication algorithm used. • `sha-256`–Secure Hash Algorithm 256 (sha-256) authentication. • `sha-384`–Secure Hash Algorithm 394 (sha-384) authentication. • `Sig key length (bits)`—Size of signature key in bits. • `Encryption`—Type of encryption algorithm used. • `aes-256-cbc`—Advanced Encryption Standard (AES) 256-bit encryption. • `aes-192-cbc`— AES192-bit encryption • `aes-128-cbc`—AES 128-bit encryption. • `3des-cbc`—3 Data Encryption Standard (DES) encryption. • `des-cbc`—DES encryption.
`Traffic statistics`	• `Input bytes`–Number of bytes received. • `Output bytes`–Number of bytes transmitted. • `Input packets`–Number of packets received. • `Output packets`–Number of packets transmitted.
`Server Info Version`	Identify the latest set of information maintained in the server.
`Server Heartbeat Interval`	Interval in seconds at which the server sends heartbeats to group members.

Table 132: show security group-vpn member kek security-associations *(Continued)*

Field Name	Field Description
`Member Heartbeat Threshold`	The heartbeat threshold configured on the group member for the IPsec VPN. If this number of heartbeats is missed on the member, the member reregisters with the server.
`Heartbeat Timeout Left`	Number of heartbeats until the heartbeat threshold is reached, at which time the member reregisters with the server. When this number reaches 0, reregistration happens within 60 seconds.
`Server Activation Delay`	Number of seconds before a group member can use a new key when the member reregisters with the server.
`Server Multicast Group`	Multicast IP address to which the server sends rekey messages.
`Server Replay Window`	Antireplay time window value in milliseconds. 0 means antireplay is disabled.
`Group Key Push sequence number`	Sequence number of the KEK SA groupkey-push message. This number is incremented with every groupkey-push message.

Sample Output

show security group-vpn member kek security-associations

```
user@host> show security group-vpn member kek security-associations
Index   Server Address  Life:sec  Initiator cookie  Responder cookie  GroupId
5824843 192.168.2.53      166       46871e26227f08f3  f0a463a4d5c3737b  1
```

Sample Output

show security group-vpn member kek security-associations detail

```
user@host> show security group-vpn member kek security-associations detail
  Index 5824843, Group Id: 1
  Group VPN Name: group1_2
  Local Gateway: 192.168.2.170, GDOI Server: 192.168.2.53
  Initiator cookie: 46871e26227f08f3, Responder cookie: f0a463a4d5c3737b
  Lifetime: Expires in 155 seconds
  Group Key Push Sequence number: 0

  Algorithms:
   Sig-hash              : hmac-md5-96
   Encryption            : 3des-cbc
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
  Stats:
      Push received          :   0
      Delete received        :   0
```

show security group-vpn member kek security-associations detail | display xml

```
user@host> show security group-vpn member kek security-associations detail | display xml

<rpc-reply xmlns:junos="http://xml.example.net/junos/15.1/junos">
    <gvpn-kek-security-associations-information junos:style="detail">
        <kek-security-associations-block>
            <security-association-index>2987691</security-association-index>
            <group-id>400</group-id>
            <group-vpn-name>gvpn400</group-vpn-name>
            <local-address>192.168.1.100</local-address>
            <server-address>192.168.1.1</server-address>
            <initiator-cookie>510f854307a03675</initiator-cookie>
            <responder-cookie>690e5f121fba6de7</responder-cookie>
            <lifetime-remaining>Expires in 23729 seconds</lifetime-remaining>
            <push-sequence-number>364</push-sequence-number>
            <ike-security-associations>
                <ike-sa-algorithms>
                    <ike-sa-authentication-algorithm>hmac-sha1-96</ike-sa-authentication-
algorithm>
                    <ike-sa-sig-key-length>2048</ike-sa-sig-key-length>
                    <ike-sa-encryption-algorithm>aes128-cbc</ike-sa-encryption-algorithm>
                </ike-sa-algorithms>
                <ike-sa-traffic-statistics>
                    <ike-sa-input-bytes>3012</ike-sa-input-bytes>
                    <ike-sa-output-bytes>252</ike-sa-output-bytes>
                    <ike-sa-input-packets>3</ike-sa-input-packets>
                    <ike-sa-output-packets>3</ike-sa-output-packets>
                </ike-sa-traffic-statistics>
            </ike-security-associations>
            <gvpn-kek-security-association-statistics>
                <kek-security-association-statistics>    Push received            :   3</kek-
security-association-statistics>
                <kek-security-association-statistics>    Delete received          :   0</kek-
security-association-statistics>
            </gvpn-kek-security-association-statistics>
        </kek-security-associations-block>
    </gvpn-kek-security-associations-information>
    <cli>
        <banner></banner>
    </cli>
</rpc-reply>
```

Release Information

Command introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

show security group-vpn member policy

IN THIS SECTION

Syntax

```
show security group-vpn member policy <vpn vpn-name> <group-id group-id>
```

Description

Show Group VPN policies. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

none — Display information for all groups.

vpn *vpn-name* — (Optional) Display policy information for the specified group name.

group-id *group-id* — (Optional) Display policy information for the specified group identifier.

Required Privilege Level

view

Output Fields

Table 133 on page 1810 lists the output fields for the `show security group-vpn member policy` command. Output fields are listed in the approximate order in which they appear.

Table 133: show security group-vpn member policy Output Fields

Field Name	Field Description
Group VPN Name	Group name.
Group Id	Group identifier.
From-zone	From zone configured for the policy.
To-zone	To zone configured for the policy.
Tunnel-id	Tunnel identifier.
Policy type	Secure, fail-open, fail-close, or exclude.
Source	IP address, port, and protocol of the source traffic.
Destination	IP address, port, and protocol of the destination traffic.

Sample Output

show security group-vpn member policy

```
user@host> show security group-vpn member policy
Group VPN Name: group1000, Group Id: 1000
From-zone: trust_1, To-zone: untrust
  Tunnel-id: 63490, Policy type: Exclude
    Source      : IP <192.168.0.0 - 192.168.255.255>, Port <0 - 65535>, Protocol <17>
    Destination : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <17>

  Tunnel-id: 49153, Policy type: Secure
    Source      : IP 192.168.0.0 - 192.168.255.255>, Port <0 - 65535>, Protocol <0>
    Destination : IP <192.0.2.0 - 192.0.2.255>, Port <0 - 65535>, Protocol <0>

  Tunnel-id: 49152, Policy type: Secure
    Source      : IP <192.0.2.0 - 192.0.2.255>, Port <0 - 65535>, Protocol <1>
    Destination : IP <192.0.2.0 - 192.0.2.255>, Port <0 - 65535>, Protocol <1>

  Tunnel-id: 63491, Policy type: Fail-open (Inactivated)
    Source      : IP 192.168.0.0 - 192.168.255.255>, Port <0 - 65535>, Protocol <17>
    Destination : IP <192.168.0.0 - 192.168.255.255>, Port <0 - 65535>, Protocol <17>

  Tunnel-id: 63489, Policy type: Fail-close
    Source      : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <0>
Destination : IP <0.0.0.0 - 255.255.255.255>, Port <0 - 65535>, Protocol <0>
```

Release Information

Command introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

show security group-vpn server ike security-associations

IN THIS SECTION

Syntax

```
show security group-vpn server ike security-associations [brief | detail] [group group-name |
group-id group-id] [index sa-index]
```

Description

Display IKE security associations (SAs). Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

- none—Display all IKE SAs for all groups.

- brief—(Optional) Display summary output.
- `detail`—(Optional) Display detailed level of output.
- `group`—(Optional) Display IKE SAs for the specified group.
- `group-id`—(Optional) Display IKE SAs for the specified group.

  An IKE SA can be used by a group member to register to multiple groups. When you specify the `group` or `group-id` options to list the IKE SAs for a specified group, all existing IKE SAs that could be used to register to the group are displayed.

- `index`—(Optional) Display information for a particular SA based on the index number of the SA. To obtain the index number for a particular SA, display the list of existing SAs by using the command with no options.

Required Privilege Level

view

Output Fields

Table 134 on page 1813 lists the output fields for the `show security group-vpn server ike security-associations` command. Output fields are listed in the approximate order in which they appear.

Table 134: show security group-vpn server ike security-associations Output Fields

Field Name	Field Description
`Index`	Index number of an SA. This number is an internally generated number you can use to display information about a single SA.
`Remote Address`	IP address of the destination peer with which the local peer communicates.

Table 134: show security group-vpn server ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`State`	State of the IKE security associations: • `DOWN`—SA has not been negotiated with the peer. • `UP`—SA has been negotiated with the peer.
`Initiator cookie`	Random number, called a cookie, which is sent to the remote node when the IKE negotiation is triggered.
`Responder cookie`	Random number generated by the remote node and sent back to the initiator as a verification that the packets were received. A cookie is aimed at protecting the computing resources from attack without spending excessive CPU resources to determine the cookie's authenticity.
`Mode`	Negotiation method agreed on by the two IPsec endpoints, or peers, used to exchange information between themselves. Each exchange type determines the number of messages and the payload types that are contained in each message. The modes, or exchange types, are • `main`—The exchange is done with six messages. This mode or exchange type encrypts the payload, protecting the identity of the neighbor. The authentication method used is displayed: preshared keys or certificate. • `aggressive`—The exchange is done with three messages. This mode or exchange type does not encrypt the payload, leaving the identity of the neighbor unprotected.
`IKE Peer`	IP address of the destination peer with which the local peer communicates.

Table 134: show security group-vpn server ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`Exchange type`	Negotiation method agreed on by the two IPsec endpoints, or peers, used to exchange information between themselves. Each exchange type determines the number of messages and the payload types that are contained in each message. The modes, or exchange types, are • `main`—The exchange is done with six messages. This mode or exchange type encrypts the payload, protecting the identity of the neighbor. The authentication method used is displayed: preshared keys or certificate. • `aggressive`—The exchange is done with three messages. This mode or exchange type does not encrypt the payload, leaving the identity of the neighbor unprotected.
`Authentication method`	Method the server uses to authenticate the source of IKE messages: • `pre-shared-keys`—Preshared key for encryption and decryption that both participants must have before beginning tunnel negotiations. • `rsa-signatures`—Digital signature, a certificate that confirms the identity of the certificate holder.
`Local`	Address of the local peer.
`Remote`	Address of the remote peer.
`Lifetime`	Number of seconds remaining until the IKE SA expires.

Table 134: show security group-vpn server ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`Algorithms`	Internet Key Exchange (IKE) algorithms used to encrypt and secure exchanges between the peers during the IPsec Phase 2 process: • `Authentication`—Type of authentication algorithm used. • `sha-256`—Secure Hash Algorithm 256 authentication. • `sha-384`—Secure Hash Algorithm 384 authentication.. • `Encryption`—Type of encryption algorithm used. • `aes-256-cbc`—Advanced Encryption Standard (AES) 256-bit encryption. • `aes-192-cbc`— AES192-bit encryption • `aes-128-cbc`—AES 128-bit encryption.
`Traffic statistics`	• `Input bytes`–Number of bytes received. • `Output bytes`–Number of bytes transmitted. • `Input packets`–Number of packets received. • `Output packets`–Number of packets transmitted.
`IPSec security associations`	• *`number`* `created`: The number of SAs created. • *`number`* `deleted`: The number of SAs deleted.

Table 134: show security group-vpn server ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`Phase 2 negotiations in progress`	Number of Phase 2 IKE negotiations in progress and status information: • `Negotiation type`—Type of Phase 2 negotiation. Junos OS currently supports quick mode. • `Message ID`—Unique identifier for a Phase 2 negotiation. • `Local identity`—Identity of the local Phase 2 negotiation. The format is id-type-name (proto-name:port-number,[0..id-data-len] = iddata-presentation) • `Remote identity`—Identity of the remote Phase 2 negotiation. The format is id-type-name (proto-name:port-number,[0..id-data-len] = iddata-presentation) • `Flags`—Notification to the key management process of the status of the IKE negotiation: • `caller notification sent`—Caller program notified about the completion of the IKE negotiation. • `waiting for done`—Negotiation is done. The library is waiting for the remote end retransmission timers to expire. • `waiting for remove`—Negotiation has failed. The library is waiting for the remote end retransmission timers to expire before removing this negotiation. • `waiting for policy manager`—Negotiation is waiting for a response from the policy manager.

Sample Output

show security group-vpn server ike security-associations

```
user@host> show security group-vpn server ike security-associations
     Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
     738879  UP     0fa7c5fdcb74669f  8c21f5d1b533010c  Aggressive     192.168.1.120
```

Sample Output

show security group-vpn server ike security-associations detail

```
user@host> show security group-vpn server ike security-associations detail
IKE peer 192.168.1.120, Index 738879, Gateway Name: gvpn
  Role: Responder, State: UP
  Initiator cookie: 0fa7c5fdcb74669f, Responder cookie: 8c21f5d1b533010c
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 192.168.1.50:848, Remote: 192.168.1.120:848
  Lifetime: Expires in 3541 seconds
  Peer ike-id: test
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha-256-128
   Encryption            : aes-256-cbc
   Pseudo random function: hmac-sha-256
   Diffie-Hellman group  : DH-group-14
  Traffic statistics:
   Input  bytes  :                  600
   Output bytes  :                  932
   Input  packets:                    4
   Output packets:                    3
  Flags: IKE SA is created
  IPSec security associations: 0 created, 0 deleted
  Phase 2 negotiations in progress: 0

Flags: IKE SA is created
```

Release Information

Command introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

show security group-vpn member ike security-associations | 1782

Group VPNv2 Overview | 758

show security group-vpn server ipsec security-associations

IN THIS SECTION

Syntax

```
show security group-vpn server ipsec security-associations [brief | detail] [group group-name |
group-id group-id]
```

Description

Display IPsec security associations (SAs). Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

- none—Display all IPsec SAs for all groups.
- `brief`—(Optional) Display summary output.
- `detail`—(Optional) Display detailed level of output.
- `group`—(Optional) Display IPsec SAs for the specified group.
- `group-id`—(Optional) Display IPsec SAs for the specified group.

Required Privilege Level

view

Output Fields

Table 135 on page 1820 lists the output fields for the `show security group-vpn server ipsec security-associations` command. Output fields are listed in the approximate order in which they appear.

Table 135: show security group-vpn server ipsec security-associations

Field Name	Field Description
`Group`	Group name.
`Group ID`	Group identifier.
`Total IPsec SAs`	The total number of IPsec SAs for each group is shown.
`IPsec SA`	Name of the SA.
`Protocol`	Protocol supported. Transport mode supports Encapsulation Security Protocol (ESP).

Table 135: show security group-vpn server ipsec security-associations *(Continued)*

Field Name	Field Description
`Algorithm`	Cryptography used to secure exchanges between peers during the IKE Phase 2 negotiations includes - An authentication algorithm used to authenticate exchanges between the peers. Options are `sha-256` and `sha-384`. - An encryption algorithm used to encrypt data traffic. Options are `aes-128-cbc`, `aes-192-cbc`, or `aes-256-cbc`.
`SPI`	Security parameter index (SPI) identifier. An SA is uniquely identified by an SPI.
`Lifetime`	The lifetime of the SA, after which it expires, expressed in seconds.
`Policy Name`	Group policy associated with the IPsec SA. The source address, destination address, source port, destination port, and protocol defined for the policy are displayed.

Sample Output

show security group-vpn server ipsec security-associations

```
user@host> show security group-vpn server ipsec security-associations
     Group: group200, Group Id: 200
       Total IPsec SAs: 1
       IPsec SA          Algorithm             SPI              Lifetime
       sa1               ESP:aes-256/sha-256    55837dfe         17
       sa1               ESP:aes-256/sha1-256   760088d          137
```

Sample Output

show security group-vpn server ipsec security-associations detail

```
user@host> show security group-vpn server ipsec security-associations detail
Group: group1, Group Id: 1
Total IPsec SAs: 10
  IPsec SA: sa1
    Protocol: ESP, Authentication: sha-256, Encryption: aes-256
    Anti-replay: D3P enabled, window size 10 milliseconds
    SPI: e68c9525
    Lifetime: Expires in 66 seconds, Activated
    Policy Name: pol1
      Source: 192.168.1.0/24
      Destination: 192.168.1.0/24
      Source Port: 0
      Destination Port: 0
      Protocol: 0
  IPsec SA: sa1
    Protocol: ESP, Authentication: sha-256, Encryption: aes-256
    Anti-replay: D3P enabled, window size 10 milliseconds
    SPI: 7ee14902
    Lifetime: Expires in 276 seconds, Activated in 36 seconds
    Rekey in 186 seconds
    Policy Name: pol1
      Source: 192.168.1.0/24
      Destination: 192.168.1.0/24
      Source Port: 0
      Destination Port: 0
      Protocol: 0
```

Release Information

Command introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

show security group-vpn member ipsec security-associations | 1792

Group VPNv2 Overview | 758

show security group-vpn server kek security-associations

IN THIS SECTION

Syntax

```
show security group-vpn server kek security-associations [brief | detail] [group group-name |
group-id group-id | index sa-index]
```

Description

Display configured server-member communications. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

- none—Display server-member communications configured for all groups.
- `brief`—(Optional) Display summary output.
- `detail`—(Optional) Display detailed output.
- `group`—(Optional) Display server-member communications configured for the specified group.
- `group-id`—(Optional) Display server-member communications configured for the specified group.
- `index`—(Optional) Display information for a particular SA based on the index number of the SA. To obtain the index number for a particular SA, display the list of existing SAs by using the command with no options.

Required Privilege Level

view

Output Fields

Table 136 on page 1824 lists the output fields for the `show security group-vpn server kek security-assocations` command. Output fields are listed in the approximate order in which they appear.

Table 136: show security group-vpn server kek security-associations Output Fields

Field Name	Field Description
`Index`	Index number of an SA. This number is an internally generated number you can use to display information about a single SA.
`Remote Address`	Identifier of the remote/peer. Because there could be multiple members, the remote address always contains the IP address 0.0.0.0.

Table 136: show security group-vpn server kek security-associations Output Fields *(Continued)*

Field Name	Field Description
`State`	State of the KEK security associations: • `DOWN`—SA is not active. • `UP`—SA is active.
`Initiator cookie`	Random number generated by the server. This is used when the server needs to push data to a member, or a member needs to reply to the server.
`Responder cookie`	Random number generated by the server. This is used when the server needs to push data to a member, or a member needs to reply to the server.
`GroupId`	Group identifier.
`KEK Peer`	IP address of the destination peer with which the local peer communicates. For KEK SAs, it always contains 0.0.0.0 which means any IP address.
`Role`	For the server, it is always initiator.
`Authentication method`	RSA is the supported authentication method.
`Local`	Address of the local peer.
`Remote`	Address of the remote peer.
`Lifetime`	Number of seconds remaining until the IKE SA expires.

Table 136: show security group-vpn server kek security-associations Output Fields *(Continued)*

Field Name	Field Description
`Algorithms`	Internet Key Exchange (IKE) algorithms used to encrypt and secure exchanges between the peers during the Phase 2 process: • `Sig-hash`—Type of authentication algorithm used. • `sha-256`—Secure Hash Algorithm 256 authentication. • `sha-384`—Secure Hash Algorithm 384 authentication. • `Encryption`—Type of encryption algorithm used. • `aes-256-cbc`—Advanced Encryption Standard (AES) 256-bit encryption. • `aes-192-cbc`— AES192-bit encryption • `aes-128-cbc`—AES 128-bit encryption.
`Traffic statistics`	• `Input bytes`–Number of bytes received. • `Output bytes`–Number of bytes transmitted. • `Input packets`–Number of packets received. • `Output packets`–Number of packets transmitted.
`Server Info Version`	Identify the latest set of information maintained in the server.
The following fields are the configured `server-member-communication` options:	
`Server Replay Window`	Antireplay time in milliseconds. This is 0 if antireplay is disabled.
`Retransmission Period`	Number of seconds between a rekey transmission and the first retransmission when there is no reply from the member.
`Number of Retransmissions`	For unicast communications, the number of times the server retransmits rekey messages to a member when there is no reply.

Table 136: show security group-vpn server kek security-associations Output Fields *(Continued)*

Field Name	Field Description
`Lifetime Seconds`	Configured lifetime, in seconds, for the KEK.
`Group Key Push sequence number`	Sequence number of the KEK SA groupkey-push message. This number is incremented with every groupkey-push message.

Sample Output

show security group-vpn server kek security-associations

```
user@host> show security group-vpn server kek security-associations
Index   Life:sec  Initiator cookie  Responder cookie  GroupId
     739031  18995     7e17278bf0a65975  0616de443d1beb77  200
```

Sample Output

show security group-vpn server kek security-associations detail

```
user@host> show security group-vpn server kek security-associations detail
Index 738879, Group Name: GROUP_ID-0001, Group Id: 1
Initiator cookie: 114e4a214891e42f, Responder cookie: 4b2848d14372e5bd
Authentication method: RSA
Lifetime: Expires in 4186 seconds, Activated
Rekey in 3614 seconds
  Algorithms:
   Sig-hash              : sha256
   Encryption            : aes256-cbc
  Traffic statistics:
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  Server Member Communication: Unicast
  Retransmission Period: 10, Number of Retransmissions: 2
  Group Key Push sequence number: 0

PUSH negotiations in progress: 0
```

Release Information

Command introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

show security group-vpn member kek security-associations | 1802

Group VPNv2 Overview | 758

show security group-vpn server registered-members

IN THIS SECTION

Syntax

```
show security group-vpn server registered-members <group group-name> <group-id group-id> <detail>
```

Description

Display currently registered group members. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

- none—Display all group members for all groups.
- brief—(Optional) Display summary output.
- detail—(Optional) Display detailed output.
- `group`—(Optional) Display group members for the specified group.
- `group-id`—(Optional) Display group members for the specified group.

Required Privilege Level

view

Output Fields

Table 137 on page 1830 lists the output fields for the `show security group-vpn server registered-members`command. Output fields are listed in the approximate order in which they appear.

Table 137: show security group—vpn server registered-members Output Fields

Field Name	Field Description
`Group`	Group name.
`Group Id`	Group identifier.
`Member Gateway`	IP address of the gateway for the group member.
`Member IP`	IP address of the group member.
`Last Update`	The last time that members registered or sent acknowledgements to the server.
`Vsys`	The root system.

Sample Output

show security group-vpn server registered-members

```
user@host> show security group-vpn server registered-members
    Group: group200, Group Id: 200
       Total number of registered members: 1
       Member Gateway                   Member IP        Last Update              Vsys
       gvpn_simpleman                   192.168.1.100    Fri Dec 20 2013 07:27:33 root
```

Sample Output

show security group-vpn server registered-members detail

```
user@host> show security group-vpn server registered-members detail
Group: group1, Group Id: 1
         Total number of registered members: 1

  Member gateway: gateway_group1_1, Member IP: 192.168.1.2, Vsys: root
  Last Update: Fri May 16 2014 03:37:17
  Stats:
      Pull Succeeded                : 321
      Pull Failed                   : 0
      Push Sent                     : 0
      Push Acknowledged             : 0
      Push Unacknowledged           : 0
```

Release Information

Command introduced in Junos OS Release 10.2.

RELATED DOCUMENTATION

clear security group-vpn server | 1688

Group VPNv2 Overview | 758

show security group-vpn server server-cluster

IN THIS SECTION

Syntax

```
show security group-vpn server server-cluster <brief> <detail> <group group-name> <group-id group-id> <peer-gateway gateway-name>
```

Description

Show information about servers in the Group VPNv2 server cluster. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

none
Display Group VPNv2 server cluster information for all groups.

brief
(Optional) Display summary output.

detail
(Optional) Display detailed output, including information about exchanges with peer servers in the cluster.

group *group-name*
(Optional) Display Group VPNv2 server cluster information for the specified group name.

group-id *group-id*
(Optional) Display Group VPNv2 server cluster information for the specified group identifier.

peer-gateway *gateway-name* — (Optional) Display Group VPNv2 server cluster information for the specified peer.

Required Privilege Level

view

Output Fields

Table 138 on page 1833 lists the output fields for the `show security group-vpn server server-cluster` command. Output fields are listed in the approximate order in which they appear.

Table 138: show security group-vpn server server-cluster Output Fields

Field Name	Field Description
Group	Group name.
Group Id	Group identifier.
Role	Role of this server in the Group VPNv2 server cluster.
Version Number	32-bit version number included in `cluster-update` exchanges and DPD probes to support anti-replay. The first `cluster-update` message sent from the root-server has version number 1. Subsequent `cluster-update` messages increment the version number by one. (Retransmit messages do not increment the version number.) Upon receipt of a `cluster-update` message, the sub-server validates the received version number. The received version number must be greater than the version number in the last received message, otherwise the message is discarded. The sub-server responds to a `cluster-update` message with an ACK message that contains the same version number as the received message. Upon receipt of the ACK message, the root-server checks that the version number is the same as in the message it sent. If the version number is valid, the exchange is considered successful. If the version number is not valid, the original message is retransmitted or the exchange is considered failed.

Table 138: show security group-vpn server server-cluster Output Fields *(Continued)*

Field Name	Field Description
Peer Gateway	Name of the peer server in the Group VPNv2 server cluster.
Peer IP	IP address of the remote peer server in the Group VPNv2 server cluster.
Role	Role of the peer server in the Group VPNv2 server cluster.
Status	Status of the peer server in the Group VPNv2 server cluster.

Sample Output

show security group-vpn server server-cluster

```
user@host> show security group-vpn server server-cluster
Group: group200, Group Id: 200
Role: Root-server, Version Number: 1,
  Peer Gateway                     Peer IP           Role                    Status
  sub_server1                      192.168.1.112     Sub-server              Active
  sub_server2                      192.168.1.113     Sub-server              Active
```

show security group-vpn server server-cluster detail

```
user@host> show security group-vpn server server-cluster detail
GGroup: group200, Group Id: 200
Role: Root-server, Version Number: 1,

Peer gateway: sub_server1,
  Peer IP: 192.168.1.112, Local IP: 192.168.1.111, VR: vr1,
  Role: Sub-server, Status: Active,
  CLUSTER-INIT send:                 0
  CLUSTER-INIT recv:                 1
  CLUSTER-INIT success:              1
  CLUSTER-INIT fail:                 0
  CLUSTER-INIT dup:                  0
  CLUSTER-INIT abort:                0
  CLUSTER-INIT timeout:              0
  CLUSTER-UPDATE send:               1
  CLUSTER-UPDATE recv:               0
  CLUSTER-UPDATE success:            1
  CLUSTER-UPDATE fail:               0
  CLUSTER-UPDATE abort:              0
  CLUSTER-UPDATE timeout:            0
  CLUSTER-UPDATE pending:            0
  CLUSTER-UPDATE max retry reached:  0
  DPD send:                          5
  DPD send fail:                     0
  DPD ACK recv:                      5
  DPD ACK invalid seqno:             0
  IPsec SA policy mismatch:          0
  IPsec SA proposal mismatch:        0
  KEK SA proposal mismatch:          0

Peer gateway: sub_server2,
  Peer IP: 192.168.1.113, Local IP: 192.168.1.111, VR: default,
  Role: Sub-server, Status: Active,
  CLUSTER-INIT send:                 0
  CLUSTER-INIT recv:                 1
  CLUSTER-INIT success:              1
  CLUSTER-INIT fail:                 0
  CLUSTER-INIT dup:                  0
  CLUSTER-INIT abort:                0
  CLUSTER-INIT timeout:              0
  CLUSTER-UPDATE send:               1
  CLUSTER-UPDATE recv:               0
  CLUSTER-UPDATE success:            1
  CLUSTER-UPDATE fail:               0
  CLUSTER-UPDATE abort:              0
  CLUSTER-UPDATE timeout:            0
  CLUSTER-UPDATE pending:            0
  CLUSTER-UPDATE max retry reached:  0
  DPD send:                          6
  DPD send fail:                     0
  DPD ACK recv:                      6
  DPD ACK invalid seqno:             0
  IPsec SA policy mismatch:          0
  IPsec SA proposal mismatch:        0
  KEK SA proposal mismatch:          0
```

Release Information

Command introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

Understanding Group VPNv2 Server Clusters | 818

show security group-vpn server statistics

IN THIS SECTION

Syntax

```
show security group-vpn server statistics <group group-name> <group-id group-id>
```

Description

Show Group VPNv2 server statistics. Group VPNv2 is supported on SRX300, SRX320, SRX340, SRX345, SRX550HM, SRX1500, SRX4100, SRX4200, and SRX4600 Series Firewalls and vSRX Virtual Firewall instances.

Options

none — Display Group VPNv2 server statistics for all groups.

group *group-name* — (Optional) Display Group VPNv2 server statistics for the specified group name.

group-id *group-id* — (Optional) Display Group VPNv2 server statistics for the specified group identifier.

Required Privilege Level

view

Output Fields

Table 139 on page 1837 lists the output fields for the `show security group-vpn server statistics` command. Output fields are listed in the approximate order in which they appear.

Table 139: show security group-vpn server statistics Output Fields

Field Name	Field Description
Group	Group name.
Group Id	Group identifier.
Stats	Server events and number of occurrences.

Sample Output

show security group-vpn server statistics

```
user@host> show security group-vpn server statistics
Group: group1, Group Id: 1
  Stats:
      Pull Succeeded                : 321
      Pull Failed                   : 0
      Pull Exceed Member Threshold  : 0
      Push Sent                     : 0
      Push Acknowledged             : 0
      Push Unacknowledged           : 0
```

Release Information

Command introduced in Junos OS Release 15.1X49-D30.

RELATED DOCUMENTATION

Group VPNv2 Overview | 758

Understanding Group VPNv2 Server Clusters | 818

show security ike active-peer

IN THIS SECTION

Syntax

```
show security ike active-peer
<peer-address>
<aaa-username username>
<brief | detail>
<debug>
local-address IP address
local-ike-id IKE ID
local-port port number (1..65535)
<fpc slot-number pic slot-number>
<ike-id IKE-ID>
<kmd-instance (all | kmd-instance-name)>
<node-local>
<pic slot-number fpc slot-number>
<port port-number peer-address>
<srg-id id-number>
routing-instance name of the local gateway routing instance
stats
<ha-link-encryption>
```

Description

Display the list of connected active users with details about the peer addresses and ports they are using.

Options

peer-address (Optional) Display details about the user with the specified peer address.

aaa-username *username*

(Optional) Display information about the user with the specified authentication, authorization, and accounting (AAA) username.

brief

(Optional) Display standard information about all users. (Default)

detail

(Optional) Display detailed information about all users.

debug

(Optional) Display debug information about all users.

local-address

Display information about the user with the specified local gateway IP address.

local-ike-id

Display information about the user with the specified local gateway IKE ID.

local-port *port-number*

Display information about users on the specified local gateway port number for specified local gateway IP address.

fpc *slot-number* pic *slot-number*

(Optional) Display information about users on the specified Flexible PIC Concentrator (FPC) slot and PIC slot.

ike-id *IKE-ID*

(Optional) Display information about the user with the specified IKE ID.

kmd-instance (all | *kmd-instance-name*)

(Optional) Display information about users in the key management process (KMD) identified by FPC *slot-number* and PIC *slot-number*.

- `all`—All KMD instances running on the Services Processing Unit (SPU).
- *`kmd-instance-name`*—Name of the KMD instance running on the SPU.

node-local

—(Optional) Display information about users for node-local tunnels in a Multinode High Availability setup.

pic *slot-number* fpc *slot-number*

(Optional) Display information about users on the specified PIC slot and FPC slot.

port *port-number peer-address*

(Optional) Display information about users on the specified port for the specified peer address.

routing-instance

Display information about users on the specified local gateway routing instance.

stats

Display detailed output along with IKE SA stats information accumulated at the peer.

ha-link-encryption

(Optional) Display information related to interchassis link (ICL) tunnel only. See "ipsec (High Availability)" on page 1544 and "show security ike active-peer ha-link-encryption (SRX5400, SRX5600, SRX5800)" on page 1845.

srg-id *number*	(Optional) Display information related to a specific services redundancy group (SRG) in a Multinode High Availability setup.

Required Privilege Level

view

Output Fields

Table 140 on page 1841 lists the output fields for the `show security ike active-peer` command. Output fields are listed in the approximate order in which they appear.

Table 140: show security ike active-peer Output Fields

Field Name	Field Description	Level of Output
`Remote Address`	IP address of the peer.	`brief`
`Port`	Port used by the peer.	All levels
`Peer IKE-ID`	IKE ID used by the peer.	All levels
`AAA username`	Username of the peer.	All levels
`Assigned IP`	IP address assigned to the peer.	`brief`
`Assigned network attributes`	Network attributes assigned to the peer can include the IP address and netmask, and DNS and WINS server addresses.	`detail`
`Previous Peer address`	IP address previously assigned to the peer.	`detail`
`Active IKE SA indexes`	Index number of the SA associated with the peer. This number is an internally generated number.	`detail`

Table 140: show security ike active-peer Output Fields *(Continued)*

Field Name	Field Description	Level of Output
`IKE SA negotiated`	Number of IKE SAs negotiated.	`detail`
`IPSec tunnels active`	Number of IPsec tunnels active.	`detail`
`IPSec Tunnel IDs`	IDs of the active IPsec tunnels.	`detail`
`DPD Config Info`	DPD configuration values.	`detail`
`DPD Statistics`	Information about DPD operations.	`detail`
`Local gateway interface`	Interface name of the local gateway.	`detail`
`Routing instance`	Name of the local gateway routing instance.	`detail`
`Local address`	IP address of the local gateway.	`detail`
`Local IKE-ID`	IKE ID used by local gateway.	`detail`

Sample Output

show security ike active-peer

```
user@host> show security ike active-peer

Remote Address    Port     Peer IKE-ID        AAA username     Assigned IP
192.168.6.136     8034     user1tac@650a      user1
192.168.80.225
```

show security ike active-peer stats

```
user@host> show security ike active-peer stats
Local gateway interface: xe-1/1/2
Routing instance: default
Local address: 192.0.2.1, Port: 500,
Local IKE-ID : device.example.net
Peer address: 198.51.100.2, Port: 500,
Peer IKE-ID : device1.example.net
AAA username: not available
Assigned network attributes:
IP Address      : 192.0.2.10 ,   netmask         : 255.255.255.0
DNS Address     : 19851.100.25 ,   DNS2 Address    : 198.51.100.26
WINS Address    : 203.0.113.25 ,   WINS2 Address   : 203.0.113.26
Assigned network attributes (IPv6):
IP Address      : :: ,   prefix          : 0
DNS Address     : 2001:db8:::ffff ,   DNS2 Address    : 2001:db8::1001
Previous Peer address   : 0.0.0.0, Port               : 0
Active IKE SA indexes   : 1
IKE SA negotiated       : 1
IPSec tunnels active    : 1, IPSec Tunnel IDs   : 500001
IKE_SA_INIT exchange stats:
Initiator stats:                                  Responder stats:
  Request Out             : 0                       Request In               : 1
  Response In             : 0                       Response Out             : 1
  Invalid KE Payload In   : 0                       Invalid KE Payload Out   : 0
  No Proposal Chosen In   : 0                       No Proposal Chosen Out   : 0
  Cookie Request In       : 0                       Cookie Request Out       : 0
  Cookie Response Out     : 0                       Cookie Response In       : 0
  Res Invalid IKE SPI     : 0                       Res DH Gen Key Fail      : 0
  Res Verify SA Fail      : 0                       Res Invalid DH Group Conf: 0
  Res IKE SA Fill Fail    : 0                       Res Get CAs Fail         : 0
  Res Verify DH Group Fail: 0                       Res Get VID Fail         : 0
  Res DH Compute Key Fail : 0                       Res DH Compute Key Fail  : 0
IKE_AUTH exchange stats:
Initiator stats:                                  Responder stats:
  Request Out             : 0                       Request In               : 1
  Response In             : 0                       Response Out             : 1
  No Proposal Chosen In   : 0                       No Proposal Chosen Out   : 0
  TS Unacceptable In      : 0                       TS Unacceptable Out      : 0
  Authentication Failed In: 0                       Authentication Failed Out: 0
IKE SA Rekey CREATE_CHILD_SA exchange stats:
```

```
Initiator stats:                                 Responder stats:
  Request Out               : 0                    Request In               : 0
  Response In               : 0                    Response Out             : 0
  No Proposal Chosen In     : 0                    No Proposal Chosen Out : 0
  Invalid KE In             : 0                    Invalid KE Out           : 0
  Res DH Compute Key Fail : 0                      Res DH Compute Key Fail: 0
  Res Verify SA Fail        : 0
  Res Fill IKE SA Fail      : 0
  Res Verify DH Group Fail: 0
IPsec SA Rekey CREATE_CHILD_SA exchange stats:
Initiator stats:                                 Responder stats:
  Request Out               : 0                    Request In               : 0
  Response In               : 0                    Response Out             : 0
  No Proposal Chosen In     : 0                    No Proposal Chosen Out : 0
  Invalid KE In             : 0                    Invalid KE Out           : 0
  TS Unacceptable In        : 0                    TS Unacceptable Out      : 0
  Res DH Compute Key Fail : 0                      Res DH Compute Key Fail: 0
  Res Verify SA Fail        : 0
  Res Verify DH Group Fail: 0
  Res Verify TS Fail        : 0
```

show security ike active-peer detail

```
user@host> show security ike active-peer detail
Local gateway interface: xe-1/1/2
Routing instance: default
Local address: 192.0.2.1, Port: 500,
Local IKE-ID : device.example.net
Peer address: 198.51.100.2, Port: 500,
Peer IKE-ID : device1.example.net
AAA username: not available
Assigned network attributes:
IP Address      : 192.0.2.10 ,   netmask         : 255.255.255.0
DNS Address     : 198.51.100.25 ,   DNS2 Address    : 198.51.100.26
WINS Address    : 203.0.113.25 ,   WINS2 Address   : 203.0.113.26
Assigned network attributes (IPv6):
IP Address      : 5000::1 ,   prefix          : 112
DNS Address     : 1000::ffff:ffff ,   DNS2 Address    : 1100::ffff:ffff

Previous Peer address   : 0.0.0.0, Port               : 0
Active IKE SA indexes   : 1
```

```
IKE SA negotiated       : 1
IPSec tunnels active    : 1, IPSec Tunnel IDs   : 500001
```

show security ike active-peer ha-link-encryption (SRX5400, SRX5600, SRX5800)

Starting in Junos OS Release 20.4R1, when you configure the high availability (HA) feature, you can use this show command to view only interchassis link tunnel details. The following command displays only interchassis link active peers and not regular active peers.

```
user@host> show security ike active-peer ha-link-encryption

Remote Address  Port     Peer IKE-ID   AAA username   Assigned IP
23.0.0.2        500      23.0.0.2      not available     0.0.0.0
```

show security ike active-peer srg-id 1

```
user@host> show security ike active-peer srg-id 1
Remote Address                      Port     Peer IKE-ID                          AAA
username                        Assigned IP
10.112.0.1                          500      10.112.0.1                                  not
available                0.0.0.0
```

show security ike active-peer node-local

```
user@host> show security ike active-peer node-local
Remote Address    Port     Peer IKE-ID                         AAA
username                        Assigned IP
6.0.0.2           500      DC=juniper, CN=r0, OU=marketing, O=juniper, L=sunnyvale,
ST=california, C=usnot available 0.0.0.0
```

```
user@host> show security ike active-peer node-local detail
Local gateway interface: xe-0/0/2.0
Routing instance: default
Local address: 4.0.0.1, Port: 500,
Local IKE-ID : DC=juniper, CN=r0, OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
Peer address: 6.0.0.2, Port: 500,
Peer IKE-ID : DC=juniper, CN=r0, OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
AAA username: not available
Assigned network attributes:
IP Address      : 0.0.0.0 ,   netmask         : 0.0.0.0
DNS Address     : 0.0.0.0 ,   DNS2 Address    : 0.0.0.0
WINS Address    : 0.0.0.0 ,   WINS2 Address   : 0.0.0.0
Assigned network attributes (IPv6):
IP Address      : :: ,   prefix          : 0
DNS Address     : :: ,   DNS2 Address    : ::

Previous Peer address   : 0.0.0.0, Port               : 0
Active IKE SA indexes   : 25
IKE SA negotiated       : 1
IPSec tunnels active    : 1, IPSec Tunnel IDs   : 500003

DPD Config Mode    : always-send
DPD Config Interval: 10
DPD Config Treshold: 3
DPD Config P1SA IDX: 25
DPD Stats Req sent: 5, DPD Stats Resp rcvd: 5
DPD Statistics          : DPD TTL                :3    DPD seq-no               :0
DPD Statistics          : DPD triggerd p1SA      :0    DPD Reserved             :0
```

Release Information

Command introduced in Junos OS Release 10.4. Support to display dead peer detection (DPD) statistics added in Junos OS Release 12.3X48-D10.

Support for the `ha-link-encryption` option added in Junos OS Release 20.4R1.

Support for the `srg-id` option added in Junos OS Release 22.4R1.

Support for the `node-local` option added in Junos OS Release 23.2R1.

RELATED DOCUMENTATION

show security ike security-associations | 1851

show security ipsec security-associations | 1899

show security ike debug-status

IN THIS SECTION

Syntax

```
show security ike debug-status
```

Description

Display debug status for currently enabled Internet Key Exchange (IKE) tracing.

Required Privilege Level

view

Output Fields

Table 141 on page 1848 lists the output fields for the `show security ike debug-status` command. Output fields are listed in the approximate order in which they appear.

Table 141: show security ike debug-status Output Fields

Field Name	Field Description
Enabled/Disabled	Status of the IKE per-tunnel tracing.
flag	Trace operation; the default is all.
level	Level of logging; the default is 7.
Local IP	Local IP address of the VPN tunnel endpoint.
Remote IP	Remote IP address of the VPN tunnel endpoint.

Sample Output

show security ike debug-status

```
user@host> show security ike debug-status
Enabled
flag: all
level: 7
Local IP: 192.0.2.1, Remote IP: 203.0.113.2
```

Release Information

Command introduced in Junos OS Release 11.4R3.

RELATED DOCUMENTATION

request security ike debug-disable | 1713

request security ike debug-enable | 1714

show security ike pre-shared-key

IN THIS SECTION

Syntax

```
show security ike pre-shared key
master-key      <master-key>
user-id          <user-id>
gateway         <gateway_name>
```

Description

Display the Internet Key Exchange (IKE) preshared key used by the Virtual Private network (VPN) gateway to authenticate the remote access user. Use either master-key or gateway option to get the master presharedkey.

Options

- `master-key` *`master-key`* —(Optional) Primary preshared key.
- `user-id` *`user-id`* —(Optional) IKE user ID value.

- `gateway` *`gateway_name`*—(Optional) Label of the VPN gateway set with master preshared key.

Required Privilege Level

view

Sample Output

show security ike pre-shared-key user-id

```
user@host> show security ike pre-shared-key user-id a@example.net master-key example
Preshared Key:3b33ec3631a561ec5a710f5d02f208033b108bb4
```

show security ike pre-shared-key gateway gateway_name user-id user-id

```
user@host> show security ike pre-shared-key gateway HUB_GW user-id user1@juniper.net
Pre-shared key: 79e4ea39f5c06834a3c4c031e37c6de24d46798a
```

Release Information

Command introduced in Junos OS Release 8.5.

`gateway` option is introduced in Junos OS Release 21.1R1.

RELATED DOCUMENTATION

policy (Security IKE) | **1579**

show security ike security-associations

IN THIS SECTION

Syntax

```
show security ike security-associations
<peer-address>
<brief | detail>
<family (inet  | inet6)>
<fpc slot-number>
<index SA-index-number>
<kmd-instance (all | kmd-instance-name)>
<node-local>
<pic slot-number>
<sa-type shortcut >
<srg-id id-number>
<ha-link-encryption>
```

Description

Display information about Internet Key Exchange security associations (IKE SAs).

Options

- none—Display standard information about existing IKE SAs, including index numbers.
- `peer-address`—(Optional) Display details about a particular SA based on the IPv4 or IPv6 address of the destination peer. This option and `index` provide the same level of output.
- `brief`—(Optional) Display standard information about all existing IKE SAs. (Default)
- `detail`—(Optional) Display detailed information about all existing IKE SAs.
- `family`—(Optional) Display IKE SAs by family. This option is used to filter the output.
  - `inet`—IPv4 address family.
  - `inet6`—IPv6 address family.
- `fpc slot-number`—(Optional) Display information about existing IKE SAs in this Flexible PIC Concentrator (FPC) slot. This option is used to filter the output.

  In a chassis cluster, when you execute the CLI command `show security ike security-associations pic <slot-number> fpc <slot-number>` in operational mode, only the primary node information about the existing IPsec SAs in the specified Flexible PIC Concentrator (FPC) slot and PIC slot is displayed.
- `index SA-index-number`—(Optional) Display information for a particular SA based on the index number of the SA. For a particular SA, display the list of existing SAs by using the command with no options. This option and `peer-address` provide the same level of output.
- `kmd-instance` —(Optional) Display information about existing IKE SAs in the key management process (in this case, it is KMD) identified by FPC *slot-number* and PIC *slot-number*. This option is used to filter the output.
  - `all`—All KMD instances running on the Services Processing Unit (SPU).
  - `kmd-instance-name`—Name of the KMD instance running on the SPU.

node-local —(Optional) Display information about IKE SAs for node-local tunnels in a Multinode High Availability setup.

- `pic slot-number` —(Optional) Display information about existing IKE SAs in this PIC slot. This option is used to filter the output.
- `sa-type`—(Optional for ADVPN) Type of SA. `shortcut` is the only option for this release.
- `ha-link-encryption`—(Optional) Display information related to interchassis link tunnel only. See "ipsec (High Availability)" on page 1544 and "show security ike security-associations ha-link-encryption (SRX5400, SRX5600, SRX5800)" on page 1868.

- `srg-id`—(Optional) Display information related to a specific services redundancy group (SRG).

Required Privilege Level

view

Output Fields

Table 142 on page 1853 lists the output fields for the `show security ike security-associations` command. Output fields are listed in the approximate order in which they appear.

Table 142: show security ike security-associations Output Fields

Field Name	Field Description
`IKE Peer or Remote Address`	IP address of the destination peer with which the local peer communicates.
`Index`	Index number of an SA. This number is an internally generated number you can use to display information about a single SA.
`Gateway Name`	Name of the IKE gateway.
`Location`	• `FPC`—Flexible PIC Concentrator (FPC) slot number. • `PIC`—PIC slot number. • `KMD-Instance`—The name of the KMD instance running on the SPU, identified by FPC *slot-number* and PIC *slot-number*. Currently, 4 KMD instances are running on each SPU, and any particular IKE negotiation is carried out by a single KMD instance.
`Role`	Part played in the IKE session. The device triggering the IKE negotiation is the initiator, and the device accepting the first IKE exchange packets is the responder.

Table 142: show security ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`State`	State of the IKE SAs: - `DOWN`—SA has not been negotiated with the peer. - `UP`—SA has been negotiated with the peer.
`Initiator cookie`	Random number, called a cookie, which is sent to the remote node when the IKE negotiation is triggered.
`Responder cookie`	Random number generated by the remote node and sent back to the initiator as a verification that the packets were received. A cookie is aimed at protecting the computing resources from attack without spending excessive CPU resources to determine the cookie's authenticity.
`Exchange type`	Negotiation method agreed on by the two IPsec endpoints, or peers, used to exchange information between one another. Each exchange type or mode determines the number of messages and the payload types that are contained in each message. The modes are: - `main`—The exchange is done with six messages. This mode encrypts the payload, protecting the identity of the neighbor. - `aggressive`—The exchange is done with three messages. This mode does not encrypt the payload, leaving the identity of the neighbor unprotected. IKEv2 protocol does not use the mode configuration for negotiation. Therefore, the mode displays the version number of the security association.
`Authentication method`	Method used to authenticate the source of IKE messages, which can be either `Pre-shared-keys` or digital certificates, such as `DSA-signatures`, `ECDSA-signatures-256`, `ECDSA-signatures-384`, or `RSA-signatures`.
`Local`	Address of the local peer.
`Remote`	Address of the remote peer.

Table 142: show security ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`Lifetime`	Number of seconds remaining until the IKE SA expires.
`Reauth Lifetime`	When enabled, number of seconds remaining until reauthentication triggers a new IKEv2 SA negotiation.
`IKE Fragmentation`	`Enabled` means that both the IKEv2 initiator and responder support message fragmentation and have negotiated the support during the IKE_SA_INIT message exchange. `Size` shows the maximum size of an IKEv2 message before it is fragmented.

Table 142: show security ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`Algorithms`	IKE algorithms used to encrypt and secure exchanges between the peers during the IPsec Phase 2 process: • `Authentication`—Type of authentication algorithm used: • `sha1`—Secure Hash Algorithm 1 authentication. • `md5`—MD5 authentication. • `Encryption`—Type of encryption algorithm used: • `aes-256-cbc`—Advanced Encryption Standard (AES) 256-bit encryption. • `aes-192-cbc`— AES192-bit encryption. • `aes-128-cbc`—AES 128-bit encryption. • `3des-cbc`—3 Data Encryption Standard (DES) encryption. • `aes-128-gcm`—Advanced Encryption Standard (AES) 256-bit encryption. • `des-cbc`—DES encryption. Starting in Junos OS Release 19.4R2, when you configure `aes-128-gcm` or `aes-256-gcm` as an encryption algorithm at the `[edit security ipsec proposalproposal-name]` hierarchy level, the authentication algorithm field of the `show security ikesecurity-associations detail` command displays the same configured encryption algorithm. • `Pseudo random function`—Function that generates highly unpredictable random numbers: `hmac-md5` or `hmac-sha1`. • `Diffie-Hellman group`—Specifies the type of Diffie-Hellman group when performing the new Diffie-Hellman exchange. It can be one of the following: • `group1`—768-bit Modular Exponential (MODP) algorithm. • `group2`—1024-bit MODP algorithm. • `group14`—2048-bit MODP group.

Table 142: show security ike security-associations Output Fields *(Continued)*

Field Name	Field Description
	- group15—3072-bit MODP algorithm. - group16—4096-bit MODP algorithm. - group19—256-bit random Elliptic Curve Groups modulo a prime (ECP group) algorithm. - group20—384-bit random ECP group algorithm. - group21—521-bit random ECP group algorithm. - group24—2048-bit MODP group with 256-bit prime order subgroup.
Traffic statistics	- Input bytes–Number of bytes received. - Output bytes–Number of bytes transmitted. - Input packets–Number of packets received. - Output packets–Number of packets transmitted. - Input fragmented packets—Number of IKEv2 fragmented packets received. - Output fragmented packets—Number of IKEv2 fragmented packets transmitted.
Flags	Notification to the key management process of the status of the IKE negotiation: - caller notification sent—Caller program notified about the completion of the IKE negotiation. - waiting for done—Negotiation is done. The library is waiting for the remote end retransmission timers to expire. - waiting for remove—Negotiation has failed. The library is waiting for the remote end retransmission timers to expire before removing this negotiation. - waiting for policy manager—Negotiation is waiting for a response from the policy manager.

Table 142: show security ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`IPSec security associations`	- *`number`* `created`: The number of SAs created. - *`number`* `deleted`: The number of SAs deleted.
`Phase 2 negotiations in progress`	Number of Phase 2 IKE negotiations in progress and status information: - `Negotiation type`—Type of Phase 2 negotiation. Junos OS currently supports quick mode. - `Message ID`—Unique identifier for a Phase 2 negotiation. - `Local identity`—Identity of the local Phase 2 negotiation. The format is *id-type-name (proto-name:port-number,[0..id-data-len] = iddata-presentation)*. - `Remote identity`—Identity of the remote Phase 2 negotiation. The format is *id-type-name (proto-name:port-number,[0..id-data-len] = iddata-presentation)*. - `Flags`—Notification to the key management process of the status of the IKE negotiation: - `caller notification sent`—Caller program notified about the completion of the IKE negotiation. - `waiting for done`—Negotiation is done. The library is waiting for the remote end retransmission timers to expire. - `waiting for remove`—Negotiation has failed. The library is waiting for the remote end retransmission timers to expire before removing this negotiation. - `waiting for policy manager`—Negotiation is waiting for a response from the policy manager.
`Local gateway interface`	Interface name of the local gateway.
`Routing instance`	Name of the local gateway routing instance.

Table 142: show security ike security-associations Output Fields *(Continued)*

Field Name	Field Description
`IPsec Tunnel IDs`	Indicates the list of child IPsec tunnel IDs

Sample Output

show security ike security-associations (IPv4)

```
user@host> show security ike security-associations
Index        Remote Address        State        Initiator cookie        Responder cookie    Mode
8        192.168.1.2        UP        3a895f8a9f620198    9040753e66d700bb    Main
Index        Remote Address        State        fInitiator cookie    Responder cookie    Mode
9        192.168.1.3         UP        5ba96hfa9f65067        70890755b65b80b        Main
```

show security ike security-associations (IPv6)

```
user@host> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
5       UP     e48efd6a444853cf  0d09c59aafb720be  Aggressive     2001:db8::1112
```

show security ike security-associations detail (SRX300, SRX320, SRX340, SRX345, and SRX550HM Devices)

```
user@host> show security ike security-associations detail
IKE peer 192.168.134.245, Index 2577565, Gateway Name: tropic
  Role: Initiator, State: UP
  Initiator cookie: b869b3424513340a, Responder cookie: 4cb3488cb19397c3
  Exchange type: Main, Authentication method: Pre-shared-keys Trusted CA group: xyz_ca_grp
  Local: 192.168.134.241:500, Remote: 192.168.134.245:500
  Local gateway interface: ge-0/0/0
  Routing instance: default
  Lifetime: Expires in 169 seconds
  Peer ike-id: 192.168.134.245
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes-128-gcm
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 1012
   Output bytes  :                 1196
   Input  packets:                    4
   Output packets:                    5
  Flags: IKE SA is created
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 0

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 192.168.134.241:500, Remote: 192.168.134.245:500
    Local identity: 192.168.134.241
    Remote identity: 192.168.134.245
    Flags: IKE SA is created
IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                  Responder stats:
    Request Out             : 1                       Request In             : 0
    Response In             : 1                       Response Out           : 0
    No Proposal Chosen In   : 0                       No Proposal Chosen Out : 0
    Invalid KE In           : 0                       Invalid KE Out         : 0
    TS Unacceptable In      : 0                       TS Unacceptable Out    : 0
    Res DH Compute Key Fail : 0                       Res DH Compute Key Fail: 0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0
```

show security ike security-associations detail (SRX5400, SRX5600, and SRX5800 Devices)

```
user@host> show security ike security-associations detail
IKE peer 2.0.0.2, Index 2068, Gateway Name: IKE_GW
  Role: Responder, State: DOWN
  Initiator cookie: aa08091f3d4f1fb6, Responder cookie: 08c89a7add5f9332
  Exchange type: IKEv2, Authentication method: Pre-shared-keys
  Local gateway interface: ge-0/0/3
  Routing instance: default
  Local: 2.0.0.1:500, Remote: 2.0.0.2:500
  Lifetime: Expires in 186 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: 2.0.0.2
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha256-128
   Encryption            : aes128-cbc
   Pseudo random function: hmac-sha256
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                   704
   Output bytes  :                  1408
   Input  packets:                     4
   Output packets:                     4
   Input  fragmented packets:       0
   Output fragmented packets:       0
  IPSec security associations: 4 created, 2 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs:  500766, 500767


Negotiation type: Quick mode, Role: Responder, Message ID: 0
Local: 2.0.0.1:500, Remote: 2.0.0.2:500
Local identity: 2.0.0.1
Remote identity: 2.0.0.2
Flags: IKE SA is created

IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                  Responder stats:
    Request Out             : 0                       Request In             : 0
    Response In             : 0                       Response Out           : 0
    No Proposal Chosen In   : 0                       No Proposal Chosen Out : 0
    Invalid KE In           : 0                       Invalid KE Out         : 0
    TS Unacceptable In      : 0                       TS Unacceptable Out    : 0
    Res DH Compute Key Fail : 0                       Res DH Compute Key Fail: 0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0
```

command-name

The "show security ike stats" on page 1871 topic lists the output fields for the `show security ike security-associations detail` command.

show security ike security-associations family inet6

```
user@host> show security ike security-associations family inet6
  IKE peer 2001:db8:1212::1112, Index 5, Gateway Name: tropic
  Role: Initiator, State: UP
  Initiator cookie: e48efd6a444853cf, Responder cookie: 0d09c59aafb720be
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 2001:db8:1212::1111:500, Remote: 2001:db8:1212::1112:500
  Lifetime: Expires in 19518 seconds
  Peer ike-id: not valid
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                 1568
   Output bytes  :                 2748
   Input  packets:                    6
   Output packets:                   23
  Flags: Caller notification sent
  IPSec security associations: 5 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 2900338624
    Local: 2001:db8:1212::1111:500, Remote: 2001:db8:1212::1112:500
    Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    Flags: Caller notification sent, Waiting for done
```

show security ike security-associations index 222075191 detail

```
user@host> show security ike security-associations index 222075191 detail
node0:
```

```
-
IKE peer 192.168.1.2, Index 222075191, Gateway Name: ZTH_HUB_GW
  Location: FPC 0, PIC 3, KMD-Instance 2
  Auto Discovery VPN:
   Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent    :    2
     Suggestions accepted:    4
     Suggestions declined:    1
  Role: Responder, State: UP
  Initiator cookie: 7b996b4c310d2424, Responder cookie: 5724c5882a212157
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.1.1:500, Remote: 192.168.1.2:500
  Lifetime: Expires in 828 seconds
  Peer ike-id: C=US, DC=example, ST=CA, L=Sunnyvale, O=example, OU=engineering, CN=cssvk36-d
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                20474
   Output bytes  :                21091
   Input  packets:                  237
   Output packets:                  237
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 192.168.1.1:500, Remote: 192.168.1.2:500
    Local identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example, OU=engineering, CN=host1
    Remote identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example, OU=engineering, CN=host2
    Flags: IKE SA is created
```

show security ike security-associations index 788674 detail

```
user@host> show security ike security-associations index 788674 detail
IKE peer 192.168.1.1, Index 788674, Gateway Name: ZTH_SPOKE_GW
  Auto Discovery VPN:
   Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:    2
     Suggestions accepted:    2
     Suggestions declined:    0
  Role: Initiator, State: UP
  Initiator cookie: 7b996b4c310d2424, Responder cookie: 5724c5882a212157
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.1.2:500, Remote: 192.168.1.1:500
  Lifetime: Expires in 734 seconds
  Peer ike-id: C=US, DC=example, ST=CA, L=Sunnyvale, O=example, OU=engineering, CN=test
  Xauth user-name: not available
  Xauth assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
  Traffic statistics:
   Input  bytes  :                22535
   Output bytes  :                21918
   Input  packets:                  256
   Output packets:                  256
  IPSec security associations: 2 created, 0 deleted
  Phase 2 negotiations in progress: 1

    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 192.168.1.2:500, Remote: 192.168.1.1:500
    Local identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example, OU=engineering, CN=host1
    Remote identity: C=US, DC=example, ST=CA, L=Sunnyvale, O=example, OU=engineering, CN=host2
    Flags: IKE SA is created
```

show security ike security-associations 192.168.1.2

```
user@host> show security ike security-associations 192.168.1.2
Index     State  Initiator cookie  Responder cookie  Mode Remote Address
   8        UP     3a895f8a9f620198  9040753e66d700bb  Main 192.168.1.2
```

show security ike security-associations fpc 6 pic 1 kmd-instance all (SRX Series Firewalls)

```
user@host> show security ike security-associations fpc 6 pic 1 kmd-instance all
Index      Remote Address  State  Initiator cookie  Responder cookie  Mode

1728053250 192.168.1.2     UP     fc959afd1070d10b  bdeb7e8c1ea99483  Main
```

show security ike security-associations detail (ADVPN Suggester, Static Tunnel)

```
user@host> show security ike security-associations detail
IKE peer 192.168.0.105, Index 13563297, Gateway Name: zth_hub_gw
  Location: FPC 0, PIC 0, KMD-Instance 1
  Auto Discovery VPN:
 Type: Static, Local Capability: Suggester, Peer Capability: Partner
   Suggester Shortcut Suggestions Statistics:
     Suggestions sent            :  12
     Suggestion response accepted:  12
     Suggestion response declined:   0
  Role: Responder, State: UP
  Initiator cookie: 4d3f4e4b2e75d727, Responder cookie: 81ab914e13cecd21
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.0.154:500, Remote: 192.168.0.105:500
  Lifetime: Expires in 26429 seconds
  Peer ike-id: DC=example, CN=host02, L=Sunnyvale, ST=CA, C=US
```

show security ike security-associations detail (ADVPN Partner, Static Tunnel)

```
user@host> show security ike security-associations detail
IKE peer 192.168.0.154, Index 4980720, Gateway Name: zth_spoke_gw
  Location: FPC 0, PIC 0, KMD-Instance 1
  Auto Discovery VPN:
 Type: Static, Local Capability: Partner, Peer Capability: Suggester
   Partner Shortcut Suggestions Statistics:
     Suggestions received:  12
     Suggestions accepted:  12
     Suggestions declined:   0
  Role: Initiator, State: UP
  Initiator cookie: 4d3f4e4b2e75d727, Responder cookie: 81ab914e13cecd21
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.0.105:500, Remote: 192.168.0.154:500
  Lifetime: Expires in 26252 seconds
  Peer ike-id: DC=example, CN=host01, OU=SBU, O=example, L=Sunnyvale, ST=CA, C=US
```

show security ike security-associations detail (ADVPN Partner, Shortcut)

```
user@host> show security ike security-associations detail
IKE peer 192.168.0.106, Index 4980737, Gateway Name: GW-ADVPN-GT-ADVPN-zth_spoke_vpn-268173323
  Location: FPC 0, PIC 0, KMD-Instance 1
  Auto Discovery VPN:
   Type: Shortcut, Local Capability: Partner, Peer Capability: Partner
  Role: Responder, State: UP
  Initiator cookie: e1ed0c655929debc, Responder cookie: 437de6ed784ba63e
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 192.168.0.105:500, Remote: 192.168.0.106:500
  Lifetime: Expires in 28796 seconds
  Peer ike-id: DC=example, CN=paulyd, L=Sunnyvale, ST=CA, C=US
```

show security ike security-associations sa-type shortcut (ADVPN)

```
user@host> show security ike security-associations sa-type shortcut
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
4980742 UP     vb56fbe694eaee5b6 064dbccbfa3b2aab  IKEv2          192.168.0.106
```

show security ike security-associations sa-type shortcut detail (ADVPN)

```
user@host> show security ike security-associations sa-type shortcut detail
IKE peer 192.168.0.106, Index 4980742, Gateway Name: GW-ADVPN-GT-ADVPN-zth_spoke_vpn-268173327
  Location: FPC 0, PIC 0, KMD-Instance 1
  Auto Discovery VPN:
   Type: Shortcut, Local Role: Partner, Peer Role: Partner
  Role: Responder, State: UP
```

show security ike security-associations detail (IKEv2 Reauthentication)

```
user@host> show security ike security-associations detail
IKE peer 10.1.2.11, Index 6009224, Gateway Name: GW
  Role: Responder, State: UP
  Initiator cookie: 2c74d14c798a9d70, Responder cookie: 83cbb49bfbcb80cb
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local: 10.1.1.11:500, Remote: 10.1.2.11:500
  Lifetime: Expires in 173 seconds
  Reauth Lifetime: Expires in 600 seconds
  Peer ike-id: vsrx@example.net
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes128-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :                 1782
   Output bytes  :                 1743
   Input  packets:                    2
```

show security ike security-associations detail (IKEv2 Fragmentation)

```
user@host> show security ike security-associations detail
IKE peer 172.24.23.157, Index 11883008, Gateway Name: routebased_s2s_gw-552_1
  Role: Responder, State: UP
  Initiator cookie: f3255e720f162e3a, Responder cookie: 17555e3ff7451841
  Exchange type: Main, Authentication method: Pre-shared-keys Trusted CA group: xyz_ca_grp
  Local: 192.168.254.1:500, Remote: 172.24.23.157:500
  Lifetime: Expires in 530 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Peer ike-id: 172.24.23.157
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-5
```

```
    Traffic statistics:
     Input  bytes  :                1004
     Output bytes  :                 756
     Input  packets:                   6
     Output packets:                   4
     Input  fragmented packets:  3
     Output fragmented packets: 3
    IPSec security associations: 1 created, 1 deleted
    Phase 2 negotiations in progress: 1

      Negotiation type: Quick mode, Role: Responder, Message ID: 0
      Local: 192.168.254.1:500, Remote: 172.24.23.157:500
      Local identity: 192.168.254.1
      Remote identity: 172.24.23.157
      Flags: IKE SA is created
```

show security ike security-associations ha-link-encryption (SRX5400, SRX5600, SRX5800)

Starting in Junos OS Release 20.4R1, when you configure the high availability (HA) feature, you can use this show command to view only interchassis link tunnel details. The following command displays only the link encryption SAs on both nodes.

```
user@host> show security ike security-associations ha-link-encryption

Index      State  Initiator cookie    Responder cookie  Mode   Remote Address
4294966287  UP    7b77b4e2fd5a87e5    ab4a398e6a28687a  IKEv2   23.0.0.2
```

show security ike security-associations srg-id

```
user@host> show security ike security-associations srg-id 1
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
16778113 UP    16d1f4efae91608c  53f234767bdd0b9b  IKEv2          10.112.0.1
```

show security ike security-associations node-local

```
user@host> show security ike security-associations node-local
Index    State  Initiator cookie  Responder cookie  Mode           Remote Address
24       UP     c982a43f5dd03bf0  c37ae96722a0e1bc  IKEv2          6.0.0.2
```

show security ike security-associations node-local detail

```
user@host> show security ike security-associations node-local
IKE peer 6.0.0.2, Index 25, Gateway Name: IKEv1_GW
  Role: Responder, State: UP
  Initiator cookie: 34b2b16c3dd35442, Responder cookie: 91fc9975f83e932d
  Exchange type: IKEv2, Authentication method: RSA-signatures
  Local gateway interface: xe-0/0/2.0
  Routing instance: default
  Local: 4.0.0.1:500, Remote: 6.0.0.2:500
  Lifetime: Expires in 1159 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: DC=juniper, CN=r0, OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
  AAA assigned IP: 0.0.0.0
  PPK-profile: None
  Algorithms:
   Authentication        : hmac-sha384-192
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha384
   Diffie-Hellman group  : DH-group-19
  Traffic statistics:
   Input  bytes  :                 3434
   Output bytes  :                 3427
   Input  packets:                   15
   Output packets:                   15
   Input  fragmented packets:       4
   Output fragmented packets:       4
  IPSec security associations: 4 created, 1 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 500003

    Negotiation type: Quick mode, Role: Responder, Message ID: 0
    Local: 4.0.0.1:500, Remote: 6.0.0.2:500
    Local identity: DC=juniper, CN=r0, OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
    Remote identity: DC=juniper, CN=r0, OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
    Flags: IKE SA is created

  IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                  Responder stats:
    Request Out             : 0                       Request In             :
0
    Response In             : 0                       Response Out           :
0
    No Proposal Chosen In   : 0                       No Proposal Chosen Out :
0
    Invalid KE In           : 0                       Invalid KE Out         :
0
    TS Unacceptable In      : 0                       TS Unacceptable Out    :
0
    Res DH Compute Key Fail : 0                       Res DH Compute Key Fail:
0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0
```

Release Information

Command introduced in Junos OS Release 8.5. Support for the `fpc`, `pic`, and `kmd-instance` options added in Junos OS Release 9.3. Support for the `family` option added in Junos OS Release 11.1. Support for Auto Discovery VPN added in Junos OS Release 12.3X48-D10. Support for IKEv2 reauthentication added in Junos OS Release 15.1X49-D60. Support for IKEv2 fragmentation added in Junos OS Release 15.1X49-D80.

Support for the `ha-link-encryption` option added in Junos OS Release 20.4R1.

Support for the `srg-id` option added in Junos OS Release 22.4R1.

Support for the `node-local` option added in Junos OS Release 23.2R1.

RELATED DOCUMENTATION

Example: Configuring a Route-Based VPN Tunnel in a User Logical Systems

show security ike stats

IN THIS SECTION

Syntax

```
show security ike stats <brief | detail>
```

Description

Display information about global IKE (Internet Key Exchange) statistics for the tunnels such as in-progress, established, and expired negotiations using IKEv2 on your SRX5000 line with SPC3 card.

Options

- **Default:** `brief`

  Displays tunnel count statistics and non-zero counters of the global IKE statistics.

`detail`

Displays all the global IKE and tunnel count statistics.

Required Privilege Level

view

Output Fields

Table 143 on page 1872 lists the output fields of total IKE SA and tunnel count statistics. Table 144 on page 1873 lists the output fields of `IKE_SA_INIT`, `IKE_AUTH`, `IKE SA Rekey CREATE_CHILD_SA`, `IPsec SA Rekey CREATE_CHILD_SA` exchanges statistics. Table 145 on page 1877 lists total IKE message failure statistics for the `show security ike stats` command. Output fields are listed in the approximate order in which they appear.

Table 143: total-IKE-SA-and-tunnel-count-statistics Output Fields

Field Name	Field Description
`Number of IKE SAs`	Number of IKE SAs currently active.
`Number of IPsec Tunnels`	Number of IPsec tunnels currently active.

Table 144: IKEV2_negotiaton_exchange_statistics

Field Name	Field Description for Output Fields of Initiator Statistics	Field Description for Output Fields of Responder Statistics
IKE_SA_INIT exchange stats	• Request Out —Number of IKE_SA_INIT request message sent by initiator. • Response In—Number of IKE_SA_INIT response message received by initiator. • Invalid KE Payload In—Number of IKE_SA_INIT INVALID_KE_PAYLOAD notification message received by initiator. • No Proposal Chosen In—Number of IKE_SA_INIT NO_PROPSAL_CHOSEN notification message received by initiator. • Cookie Request In—Number of IKE_SA_INIT cookie request notification message received by initiator. • Cookie Response Out—Number of IKE_SA_INIT cookie response notification message sent by responder. • Res Invalid IKE SPI—Number of IKE_SA_INIT response message containing invalid SPI received by initiator. • Res Verify SA Fail—Number of IKE_SA_INIT response message processing failed during verification of peer SA at initiator. • Res IKE SA Fill Fail—Number of IKE_SA_INIT response message processing failed during verification of IKE SA fill operation at initiator.	• Request In—Number of IKE_SA_INIT request message received by responder. • Response Out—Number of IKE_SA_INIT response message sent by responder. • Invalid KE Payload Out—Number of IKE_SA_INIT INVALID_KE_PAYLOAD notification message sent by responder. • No Proposal Chosen Out—Number of IKE_SA_INIT NO_PROPSAL_CHOSEN notification message sent by responder. • Cookie Request Out—Number of IKE_SA_INIT cookie request notification message sent by responder. • Cookie Response In—Number of IKE_SA_INIT cookie response notification message received by responder. • Res DH Gen Key Fail—Number of IKE_SA_INIT response message processing failed during Diffie-Hellman generate key at responder. • Res Invalid DH Group Conf—Number of IKE_SA_INIT response message processing failed due to invalid Diffie-Hellman group configured at responder. • Res Get CAs Fail—Number of IKE_SA_INIT response message processing failed during get CAs operation at responder. • Res Get VID Fail—Number of IKE_SA_INIT response message processing failed during get vendor ID request operation at responder.

Table 144: IKEV2_negotiaton_exchange_statistics *(Continued)*

Field Name	Field Description for Output Fields of Initiator Statistics	Field Description for Output Fields of Responder Statistics
	• Res Verify DH Group Fail—Number of IKE_SA_INIT response message processing failed during verification of Diffie-Hellman group at initiator. • Res DH Compute Key Fail—Number of IKE_SA_INIT response message processing failed during verification of Diffie-Hellman compute key at initiator.	• Res DH Compute Key Fail—Number of IKE_SA_INIT response message processing failed during Diffie-Hellman compute key at responder.
IKE_AUTH exchange stats	• Request Out—Number of IKE_AUTH request message sent by initiator. • Response In—Number of IKE_AUTH response message received by initiator. • No Proposal Chosen In—Number of IKE_AUTH NO_PROPSAL_CHOSEN notification message received by initiator. • TS Unacceptable In—Number of IKE_AUTH TS_UNACCEPTABLE notification message received by initiator. • Authentication Failed In—Number of IKE_AUTH AUTHENTICATION_FAILED notification message received by initiator.	• Request In—Number of IKE_AUTH request message received by responder. • Response Out—Number of IKE_AUTH response message sent by responder. • No Proposal Chosen Out—Number of IKE_AUTH NO_PROPSAL_CHOSEN notification message sent by responder. • TS Unacceptable out—Number of IKE_AUTH TS_UNACCEPTABLE notification message sent by responder. • Authentication Failed Out—Number of IKE_AUTH AUTHENTICATION_FAILED notification message sent by responder.

Table 144: IKEV2_negotiaton_exchange_statistics *(Continued)*

Field Name	Field Description for Output Fields of Initiator Statistics	Field Description for Output Fields of Responder Statistics
IKE SA Rekey CREATE_CHILD_SA exchange stats	• Request Out—Number of IKE SA rekey CREATE_CHILD_SA request message sent by initiator. • Response In—Number of IKE SA rekey CREATE_CHILD_SA response message received by initiator. • No Proposal Chosen In—Number of IKE SA rekey CREATE_CHILD_SA NO_PROPSAL_CHOSEN notification message received by initiator. • Invalid KE In—Number of IKE SA rekey CREATE_CHILD_SA INVALID_KE_PAYLOAD notification message received by initiator. • Res DH Compute Key Fail—Number of IKE SA rekey CREATE_CHILD_SA response message processing failed during verification of Diffie-Hellman compute key at initiator. • Res Verify SA Fail—Number of IKE SA rekey CREATE_CHILD_SA response message processing failed during verification of peer SA failed at initiator. • Res Fill IKE SA Fail—Number of IKE SA rekey CREATE_CHILD_SA response message processing failed during IKE SA fill operation at initiator. • Res Verify DH Group Fail—Number of IKE SA rekey CREATE_CHILD_SA response message processing failed during verification of Diffie-Hellman group at initiator.	• Request In—Number of IKE SA rekey CREATE_CHILD_SA request message received by responder. • Response Out—Number of IKE SA rekey CREATE_CHILD_SA response message sent by responder. • No Proposal Chosen Out—Number of IKE SA rekey CREATE_CHILD_SA NO_PROPSAL_CHOSEN notification message sent by responder. • Invalid KE Out—Number of IKE SA rekey CREATE_CHILD_SA INVALID_KE_PAYLOAD notification message sent by responder. • Res DH Compute Key Fail—Number of IKE SA rekey CREATE_CHILD_SA response message processing failed during Diffie-Hellman compute key at responder.

Table 144: IKEV2_negotiaton_exchange_statistics *(Continued)*

Field Name	Field Description for Output Fields of Initiator Statistics	Field Description for Output Fields of Responder Statistics
IPsec SA Rekey CREATE_CHILD_SA exchange stats	• Request Out—Number of IPsec SA rekey CREATE_CHILD_SA request message sent by initiator. • Response In—Number of IPsec SA rekey CREATE_CHILD_SA response message received by initiator. • No Proposal Chosen In—Number of IPsec SA rekey CREATE_CHILD_SA NO_PROPSAL_CHOSEN notification message received by initiator. • Invalid KE In—Number of IPsec SA rekey CREATE_CHILD_SA INVALID_KE_PAYLOAD notification message received by initiator. • TS Unacceptable In—Number of IPsec SA rekey CREATE_CHILD_SA TS_UNACCEPTABLE notification message received by initiator. • Res DH Compute Key Fail—Number of IPsec SA rekey CREATE_CHILD_SA response message processing failed during verification of Diffie-Hellman compute key at initiator. • Res Verify SA Fail—Number of IPsec SA rekey CREATE_CHILD_SA response message processing failed during verification of peer SA at initiator. • Res Verify DH Group Fail—Number of IPsec SA rekey CREATE_CHILD_SA response message processing failed during verification of Diffie-Hellman group at initiator.	• Request In—Number of IPsec SA rekey CREATE_CHILD_SA request message received by responder. • Response Out—Number of IPsec SA rekey CREATE_CHILD_SA response message sent by responder. • No Proposal Chosen Out—Number of IPsec SA rekey CREATE_CHILD_SA NO_PROPSAL_CHOSEN notification message sent by responder. • Invalid KE Out—Number of IPsec SA rekey CREATE_CHILD_SA INVALID_KE_PAYLOAD notification message sent by responder. • TS Unacceptable Out—Number of IPsec SA rekey CREATE_CHILD_SA TS_UNACCEPTABLE notification message sent by responder. • Res DH Compute Key Fail—Number of IPsec SA rekey CREATE_CHILD_SA response message processing failed during Diffie-Hellman compute key at responder.

Table 144: IKEV2_negotiaton_exchange_statistics *(Continued)*

Field Name	Field Description for Output Fields of Initiator Statistics	Field Description for Output Fields of Responder Statistics
	• `Res Verify TS Fail`—Number of IPsec SA rekey `CREATE_CHILD_SA` response message processing failed during verification of TS at initiator.	

Table 145: IKEv2_negotiation_message_failure_statistics

Field Name	Field Description
`Discarded`	The total number of discarded messages.
`Integrity fail`	The total number of messages with integrity check failure.
`Invalid exchange type`	The total number of messages with invalid exchange type failure.
`Disorder`	The total number of messages failure due to disorder.
`ID error`	The total number of messages with ID error.
`Invalid SPI`	The total number of messages with invalid SPI failure.
`Invalid length`	The total number of messages with invalid length failure.

Sample Output

show security ike stats brief

```
user@host> show security ike stats brief
Total IKE SA and Tunnel Count Statistics:
  Number of IKE SAs: 2             Number of IPsec Tunnels: 2

IKE_SA_INIT exchange stats:
 Initiator stats:                                  Responder stats:
                                                    Request In               : 4
                                                    Response Out             : 4

IKE_AUTH exchange stats:
 Initiator stats:                                  Responder stats:
                                                    Request In               : 4
                                                    Response Out             : 4

IKE SA Rekey CREATE_CHILD_SA exchange stats:
 Initiator stats:                                  Responder stats:
  Request Out             : 1                       Request In             : 1
  Response In             : 1                       Response Out           : 1

IPsec SA Rekey CREATE_CHILD_SA exchange stats:
 Initiator stats:                                  Responder stats:
  Request Out             : 1537
  Response In             : 1537
```

Sample Output

show security ike stats detail

```
user@host> show security ike stats detail
Total IKE SA and Tunnel Count Statistics:
  Number of IKE SAs: 2             Number of IPsec Tunnels: 2

IKE_SA_INIT exchange stats:
 Initiator stats:                                  Responder stats:
  Request Out             : 0                       Request In               : 4
  Response In             : 0                       Response Out             : 4
  Invalid KE Payload In   : 0                       Invalid KE Payload Out   : 0
  No Proposal Chosen In   : 0                       No Proposal Chosen Out   : 0
  Cookie Request In       : 0                       Cookie Request Out       : 0
  Cookie Response Out     : 0                       Cookie Response In       : 0
  Res Invalid IKE SPI     : 0                       Res DH Gen Key Fail      : 0
  Res Verify SA Fail      : 0                       Res Invalid DH Group Conf: 0
  Res IKE SA Fill Fail    : 0                       Res Get CAs Fail         : 0
  Res Verify DH Group Fail: 0                       Res Get VID Fail         : 0
  Res DH Compute Key Fail : 0                       Res DH Compute Key Fail  : 0

IKE_AUTH exchange stats:
 Initiator stats:                                  Responder stats:
  Request Out             : 0                       Request In               : 4
  Response In             : 0                       Response Out             : 4
  No Proposal Chosen In   : 0                       No Proposal Chosen Out   : 0
  TS Unacceptable In      : 0                       TS Unacceptable Out      : 0
  Authentication Failed In: 0                       Authentication Failed Out: 0

IKE SA Rekey CREATE_CHILD_SA exchange stats:
 Initiator stats:                                  Responder stats:
  Request Out             : 1                       Request In             : 1
  Response In             : 1                       Response Out           : 1
  No Proposal Chosen In   : 0                       No Proposal Chosen Out : 0
  Invalid KE In           : 0                       Invalid KE Out         : 0
  Res DH Compute Key Fail : 0                       Res DH Compute Key Fail: 0
  Res Verify SA Fail      : 0
  Res Fill IKE SA Fail    : 0
  Res Verify DH Group Fail: 0

IPsec SA Rekey CREATE_CHILD_SA exchange stats:
 Initiator stats:                                  Responder stats:
  Request Out             : 1537                    Request In             : 0
  Response In             : 1537                    Response Out           : 0
  No Proposal Chosen In   : 0                       No Proposal Chosen Out : 0
  Invalid KE In           : 0                       Invalid KE Out         : 0
  TS Unacceptable In      : 0                       TS Unacceptable Out    : 0
  Res DH Compute Key Fail : 0                       Res DH Compute Key Fail: 0
  Res Verify SA Fail      : 0
  Res Verify DH Group Fail: 0
  Res Verify TS Fail      : 0

Total IKE message failure stats:
  Discarded            : 0                          ID error      : 0
  Integrity fail       : 0                          Invalid SPI   : 0
  Invalid exchange type: 0                          Invalid length: 0
  Disorder             : 0
```

Release Information

Command introduced in Junos OS Release 19.4R1.

CLI options `brief` and `detail` are introduced in Junos OS Release 20.1R1.

RELATED DOCUMENTATION

Phase 1 of IKE Tunnel Negotiation | 12

Phase 2 of IKE Tunnel Negotiation | 14

show security ike tunnel-map

IN THIS SECTION

Syntax

```
show security ike tunnel-map (<brief | summary>) <fpc slot-number> <kmd-instance (all | kmd-
instance-name)> <pic slot-number>
```

Description

Display the tunnel mapping on different Services Processing Units (SPUs) for site-to-site and manual VPNs. You can insert an SPC on a device in a chassis cluster without disrupting traffic on the existing VPN tunnels. After inserting the SPC, you can view the tunnel mapping using this command. This feature is supported only on SRX5400, SRX5600, and SRX5800 Series Firewalls and vSRX Virtual Firewall instances.

Options

brief

Display standard information about all existing IKE SAs. This is the default.

fpc *slot-number*

Display information about existing IKE SAs in the specified Flexible PIC Concentrator (FPC) slot.

kmd-instance (all | *kmd-instance-name*)

(Optional) Display information about existing IKE SAs in the key management process (KMD) identified by FPC *slot-number* and PIC *slot-number*. This option is used to filter the output. You can specify one of the following options:

- all—All KMD instances running on the Services Processing Unit (SPU).
- *kmd-instance-name*—Name of the KMD instance running on the SPU.

pic *slot-number*

Display information about existing IKE SAs in the specified PIC slot.

summary

Display the tunnel-mapping load on each SPU. The load is the number of times an SPU has been chosen as an anchor SPU. For site-to-site VPNs, the load should be equal to the number of gateways mapped to an SPU.

Required Privilege Level

view

Output Fields

Table 146 on page 1882 lists the output fields for the `show security ike tunnel-map` command. Output fields are listed in the approximate order in which they appear.

Table 146: show security ike tunnel-map Output Fields

Field Name	Field Descripton
Gateway ID	Gateway identifier. This is a nondeterministic number that is constant as long as the configuration is present. This number does not appear in any other outputs.
Gateway Name	Name of the IKE gateway.
FPC	FPC slot number.
PIC	PIC slot number.
IKED Instance	IKE process instance identifier.
SPU Load	Number of times an SPU has been chosen as an anchor SPU.

Sample Output

show security ike tunnel-map

```
user@host> show security ike tunnel-map
Gateway ID   Gateway Name FPC  PIC  IKED Instance
    2        ike_gw1       4    0       1
    3        ike_gw2       7    0       1
    4        ike_gw3       7    0       2
    5        ike_gw4       4    0       2
```

show security ike tunnel-map brief

```
user@host> show security ike tunnel-map brief
Gateway ID   Gateway Name FPC  PIC  IKED Instance
    2        gw-01         1    0       1
    3        LAN_1         1    0       2
    4        LAN_2         1    0       1
    5        LAN_3         1    0       2
    6        LAN_4         1    0       1
```

show security ike tunnel-map fpc 1 pic 0

```
user@host> run show security ike tunnel-map fpc 1 pic 0
Gateway ID   Gateway Name FPC  PIC  IKED Instance
    2        gw-01         1    0       1
    3        LAN_1         1    0       2
    4        LAN_2         1    0       1
    5        LAN_3         1    0       2
    6        LAN_4         1    0       1
```

show security ike tunnel-map kmd-instance kmd1

```
user@host> show security ike tunnel-map kmd-instance kmd1
Gateway ID   Gateway Name FPC  PIC  IKED Instance
    2        gw-01         1    0       1
    4        LAN_2         1    0       1
    6        LAN_4         1    0       1
```

show security ike tunnel-map kmd-instance all

```
user@host> show security ike tunnel-map kmd-instance all
Gateway ID   Gateway Name FPC  PIC  IKED Instance
    2        gw-01         1    0       1
    3        LAN_1         1    0       2
    4        LAN_2         1    0       1
    5        LAN_3         1    0       2
    6        LAN_4         1    0       1
```

show security ike tunnel-map summary

```
user@host> show security ike tunnel-map summary
FPC  PIC  SPU Load
1    0      5
```

Release Information

Command introduced in Junos OS Release 12.1X44-D10.

RELATED DOCUMENTATION

VPN Support for Inserting Services Processing Cards | 175

show security ipsec control-plane-security-associations

IN THIS SECTION

Syntax

```
show security ipsec control-plane-security-associations
<brief | detail>
<sa-name sa-name>
```

Description

Display information about manual IPsec security associations (SAs) applied to OSPF or OSPFv3 interfaces or virtual links.

Options

- `brief` | `detail`—(Optional) Display the specified level of output.
- `sa-name` *sa-name*—Name of the manual SA.

Required Privilege Level

view

Output Fields

Table 147 on page 1885 lists the output fields for the `show security ipsec control-plane-security-associations` command. Output fields are listed in the approximate order in which they appear.

Table 147: show security ipsec control-plane-security-associations Output Fields

Field Name	Field Description
`Name`	Name of the SA.

Table 147: show security ipsec control-plane-security-associations Output Fields *(Continued)*

Field Name	Field Description
`Algorithm`	IPsec protocol followed by encryption algorithm and authentication algorithm.
`SPI`	SPI value.
`Total active security-associations`	Total number of active manual SAs for application to OSPF or OSPFv3 interfaces or virtual links.

Sample Output

show security ipsec control-plane-security-associations

```
user@host> show security ipsec control-plane-security-associations
Name      Algorithm       SPI
test_sa   ESP:3des/md5   3e8
test_sa   ESP:3des/md5   3e8
test_sa2  ESP:3des/sha1  7d1
test_sa2  ESP:3des/sha1  7d1
Total active security-associations: 2
```

show security ipsec control-plane-security-associations sa-name

```
user@host> show security ipsec control-plane-security-associations sa-name test_sa
Name      Algorithm       SPI
test_sa   ESP:3des/md5    3e8
test_sa   ESP:3des/md5    3e8
Total active security-associations: 1
```

show security ipsec control-plane-security-associations detail

```
user@host> show security ipsec control-plane-security-associations detail
Direction: inbound, SA Name: test_sa,
Protocol: ESP:, Authentication: md5
SPI: 3e8, AUX-SPI: 0,
Mode: transport, Type: manual,
ID: 1,

Direction: outbound, SA Name: test_sa,
Protocol: ESP:, Authentication: md5
SPI: 3e8, AUX-SPI: 0,
Mode: transport, Type: manual,
ID: 2,

Direction: inbound, SA Name: test_sa2,
Protocol: ESP:, Authentication: sha1
SPI: 7d1, AUX-SPI: 0,
Mode: transport, Type: manual,
ID: 3,

Direction: outbound, SA Name: test_sa2,
Protocol: ESP:, Authentication: sha1
SPI: 7d1, AUX-SPI: 0,
Mode: transport, Type: manual,
ID: 4,
```

Release Information

Command introduced in Junos OS Release 12.1X46-D20.

RELATED DOCUMENTATION

Understanding OSPF and OSPFv3 Authentication on SRX Series Firewalls | 199

show security ipsec inactive-tunnels

IN THIS SECTION

Syntax

```
show security ipsec inactive-tunnels
brief | detail
family (inet  | inet6)
fpc slot-number
index index-number
kmd-instance (all | kmd-instance-name)
node-local
pic slot-number
srg-id id-number
sa-type shortcut
vpn-name vpn-name
```

Description

Display security information about the inactive tunnel.

Options

- none—Display information about all inactive tunnels.
- `brief` | `detail`—(Optional) Display the specified level of output.
- `family`—(Optional) Display the inactive tunnel by family. This option is used to filter the output.
  - `inet`—IPv4 address family.
  - `inet6`—IPv6 address family.
- `fpc` *`slot-number`*—(Optional) Display information about inactive tunnels in the Flexible PIC Concentrator (FPC) slot.
- `index` *`index-number`*—(Optional) Display detailed information about the specified inactive tunnel identified by this index number. For a list of all inactive tunnels with their index numbers, use the command with no options.
- `kmd-instance` —(Optional) Display information about inactive tunnels in the key management process (in this case, it is KMD) identified by FPC *slot-number* and PIC *slot-number*.
  - `all`—All KMD instances running on the Services Processing Unit (SPU).
  - *`kmd-instance-name`*—Name of the KMD instance running on the SPU.
- `node-local`—(Optional) Display information about inactive tunnels for node-local tunnels in a Multinode High Availability setup.
- `pic` *`slot-number`*—Display information about inactive tunnels in the PIC slot.
- `sa-type`—(Optional for ADVPN) Type of SA. `shortcut` is the only option for this release.
- `vpn-name` *`vpn-name`*—(Optional) Name of the VPN.
- `srg-id`*`id-number`*—(Optional) Display information related to a specific services redundancy group (SRG) in a Multinode High Availability setup.

The `fpc` *`slot-number`*, `kmd-instance (all |` *`kmd-instance-name`*`)`, and `pic` *`slot-number`* parameters apply to SRX5600 and SRX5800 devices only.

Required Privilege Level

view

Output Fields

Table 148 on page 1890 lists the output fields for the `show security ipsec inactive-tunnels` command. Output fields are listed in the approximate order in which they appear.

Table 148: show security ipsec inactive-tunnels Output Fields

Field Name	Field Description
`Total inactive tunnels`	Total number of inactive IPsec tunnels.
`Total inactive tunnels which establish immediately`	Total number of inactive IPsec tunnels that can establish a session immediately.
`ID`	Identification number of the inactive tunnel. You can use this number to get more information about the inactive tunnel.
`Gateway`	IP address of the remote gateway.
`Port`	If Network Address Translation (NAT) is used, this value is 4500. Otherwise, it is the standard IKE port, 500.
`Def-Del#`	Number of deferred deletions of a dial-up IPsec VPN.
`Virtual system`	Virtual system to which the VPN belongs.
`VPN name`	Name of the IPsec VPN.
`Local gateway`	Gateway address of the local system.
`Remote gateway`	Gateway address of the remote system.

Table 148: show security ipsec inactive-tunnels Output Fields *(Continued)*

Field Name	Field Description
`Traffic Selector Name`	For IPsec running KMD process - • Displays the name only when traffic selector is configured. • Doesn’t display anything if traffic selector is not configured. For IPsec running IKED-NG process, by default - • Displays the name when traffic selector is configured. • Displays the name as *default_proxyid* when `proxy-identity` is configured. • Displays the name as *default_any_any* when traffic selector is not configured. See "show security ipsec inactive-tunnels detail" on page 1893, for more details.
`Local identity`	Identity of the local peer so that its partner destination gateway can communicate with it. The value is specified as an IP address, fully qualified domain name, e-mail address, or distinguished name (DN). Displays `proxy-identity` when configured for IPsec running either KMD or IKED-NG process. Displays *0.0.0.0* when `proxy-identity` is not configured. See "show security ipsec inactive-tunnels detail" on page 1893, for more details.
`Remote identity`	Identity of the destination peer gateway. The value is specified as an IP address, fully qualified domain name, e-mail address, or distinguished name (DN). Displays `proxy-identity` when configured for IPsec running either KMD or IKED-NG process. Displays *0.0.0.0* when `proxy-identity` is not configured. See "show security ipsec inactive-tunnels detail" on page 1893, for more details.

Table 148: show security ipsec inactive-tunnels Output Fields *(Continued)*

Field Name	Field Description
`Version`	Version of IKE.
`Passive Mode Tunneling`	IPsec tunneling of malformed packets; enabled if set or disabled if not set.
`DF-bit`	State of the don't fragment bit: `set` or `clear`.
`Bind-interface`	The tunnel interface to which the route-based VPN is bound.
`Policy-name`	Name of the applicable policy.
`Tunnel Down Reason`	Reason for which the tunnel is inactive.
`Tunnel events`	Tunnel event and the number of times the event has occurred. See Tunnel Events for descriptions of tunnel events and the action you can take.

Sample Output

show security ipsec inactive-tunnels

```
user@host> show security ipsec inactive-tunnels
Total inactive tunnels: 1
  Total inactive tunnels with establish immediately: 0
  ID     Gateway     Port  Tunnel down reason
  131073 192.168.1.2  500   Phase1 proposal mismatch detected
```

show security ipsec inactive-tunnels detail

For IPsec running KMD process, when both proxy-identity and traffic-selector are not configured.

```
user@host> show security ipsec inactive-tunnels detail
ID: 131073 Virtual-system: root, VPN Name: vpn1
  Local Gateway: 192.12.0.20, Remote Gateway: 192.12.0.10
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
```

For IPsec running KMD process, when proxy-identity is configured.

```
user@host> show security ipsec inactive-tunnels detail
ID: 131074 Virtual-system: root, VPN Name: vpn1
  Local Gateway: 192.12.0.20, Remote Gateway: 192.12.0.10
  Local Identity: ipv4_subnet(any:0,[0..7]=1.0.0.0/8)
  Remote Identity: ipv4_subnet(any:0,[0..7]=4.0.0.0/8)
```

For IPsec running KMD process, when traffic-selector is configured.

```
user@host> show security ipsec inactive-tunnels detail
ID: 67108865 Virtual-system: root, VPN Name: vpn1
  Local Gateway: 192.12.0.20, Remote Gateway: 192.12.0.10
  Traffic Selector Name: ts1
  Local Identity: ipv4(1.0.0.0-1.255.255.255)
  Remote Identity: ipv4(4.0.0.0-4.255.255.255)
```

For IPsec running IKED-NG process, when traffic-selector is configured.

```
user@host> show security ipsec inactive-tunnels detail
ID: 105 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 2.0.0.1, Remote Gateway: 2.0.0.5
  Traffic Selector Name: ts1
  Local Identity: ipv4(1.0.0.0-1.255.255.255)
  Remote Identity: ipv4(4.0.0.0-4.255.255.255)
```

For IPsec running IKED-NG process, when `traffic-selector` is not configured.

```
user@host> show security ipsec inactive-tunnels detail
ID: 107 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 2.0.0.1, Remote Gateway: 2.0.0.5
  Traffic Selector Name: default_any_any_v4
  Local Identity: ipv4(0.0.0.0-255.255.255.255)
  Remote Identity: ipv4(0.0.0.0-255.255.255.255)
```

For IPsec running IKED-NG process, when `proxy-identity` is configured.

```
user@host> show security ipsec inactive-tunnels detail
ID: 110 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 2.0.0.1, Remote Gateway: 2.0.0.5
  Traffic Selector Name: default_proxyid
  Local Identity: ipv4(1.0.0.0-1.255.255.255)
  Remote Identity: ipv4(4.0.0.0-4.255.255.255)
```

show security ipsec inactive-tunnels index 131073

```
user@host> show security ipsec inactive-tunnels index 131073
ID: 131073 Virtual-system: root, VPN Name: vpn1
  Local Gateway: 192.168.1.100, Remote Gateway: 192.168.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.0
  Port: 500, Nego#: 2, Fail#: 0, Def-Del#: 0 Flag: 600a29
  Tunnel events:
    Wed Jul 16 2014 06:18:02 +0800: User cleared IPSec SA from CLI (1 times)
    Wed Jul 16 2014 06:17:58 +0800: IPSec SA negotiation successfully completed (1 times)
    Wed Jul 16 2014 06:17:54 +0800: User cleared IPSec SA from CLI (1 times)
    Wed Jul 16 2014 06:16:58 +0800: IPSec SA negotiation successfully completed (1 times)
    Wed Jul 16 2014 06:16:58 +0800: Bind interface's address received. Information updated (1
times)
    Wed Jul 16 2014 06:16:58 +0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Wed Jul 16 2014 06:16:58 +0800: External interface's address received. Information updated
(1 times)
```

```
    Wed Jul 16 2014 06:16:58 +0800: Bind interface's zone received. Information updated (1 times)
    Wed Jul 16 2014 06:16:58 +0800: IKE SA negotiation successfully completed (1 times)
```

show security ipsec inactive-tunnels sa-type shortcut

```
user@host> show security ipsec inactive-tunnels sa-type shortcut
  Total inactive tunnels: 1
  Total inactive tunnels with establish immediately: 0
  ID     Port  Nego#  Fail#  Flag      Gateway          Tunnel Down Reason
  268173322 500 0     0      40608aa9  192.168.0.105       Cleared via CLI
```

show security ipsec inactive-tunnels with passive mode tunneling

```
user@host>show security ipsec inactive-tunnels
  ID: 6 Virtual-system: root, VPN Name: vpn2
  Local Gateway: 10.0.0.2, Remote Gateway: 30.0.0.2
  Traffic Selector Name: ts2
  Local Identity: ipv4(50.0.1.0-50.0.1.255)
  Remote Identity: ipv4(140.0.1.0-140.0.1.255)
  Version: IKEv2
  Passive mode tunneling: Disabled
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: ipsec_policy
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
```

show security ipsec inactive-tunnels node-local

```
user@host>show security ipsec inactive-tunnels node-local
  Total inactive tunnels: 0
  Total inactive tunnels with establish immediately: 0
```

Release Information

Command introduced in Junos OS Release 11.4R3.

Support for the `passive-mode-tunneling` option on MX-SPC3 is introduced in Junos OS Release 23.1R1.

Support for the `node-local` option is added in Junos OS Release 23.2R1.

RELATED DOCUMENTATION

show security ipsec security-associations

show security ipsec next-hop-tunnels

IN THIS SECTION

Syntax

```
show security ipsec next-hop-tunnels {
    family (inet | inet6);
    index;
    interface-name;
}
```

Description

Display security information about the secure tunnel interface.

Options

family	Display IPSec next-hop-tunnel entries by family.
index	Index of security association. - Range: - 1 through 4294967295
inet	Displays IPv4 protocol parameters.
inet6	Displays IPv6 protocol parameters.
interface-name	Name of the secure tunnel logical interface.

Required Privilege Level

view

Output Fields

Table 149 on page 1897 lists the output fields for the `show security ipsec next-hop-tunnels` command. Output fields are listed in the approximate order in which they appear.

Table 149: show security ipsec next-hop-tunnels Output Fields

Field Name	Field Description
`Next-hop gateway`	IP address of the next gateway.
`Interface`	Name of the secure tunnel logical interface.
`IPsec VPN name`	Name of the IPsec VPN tunnel.

Table 149: show security ipsec next-hop-tunnels Output Fields *(Continued)*

Field Name	Field Description
`Flag`	- `Static`—IP address manually configured. - `Auto`—IP address obtained from the remote peer automatically.

Sample Output

show security ipsec next-hop-tunnels family inet

```
user@host> show security ipsec next-hop-tunnels inet
Next-hop gateway     interface   IPsec VPN name                    Flag
192.168.1.2          st0.0       autokey                           Static
192.168.1.3          st0.0       pbd-4-6                           Auto
```

show security ipsec next-hop-tunnels family inet6

```
user@host> show security ipsec next-hop-tunnels family inet6
Next-hop gateway                interface   IPSec VPN name            Flag
2001:db8::2                     st0.1       IPSEC_VPNA_1              Auto
2001:db8::3                     st0.1       IPSEC_VPNA_1              Auto
2001:fe80::5668:ad10:fcd8:59db  st0.1       IPSEC_VPNA_1              Auto
2001:fe80::5668:ad10:fcd8:5aa5  st0.1       IPSEC_VPNA_1              Auto
```

Release Information

Command introduced in Junos OS Release 8.5.

The `family inet6` option is introduced in Junos OS Release 18.1R1.

RELATED DOCUMENTATION

show security ipsec security-associations | 1899

show security ipsec security-associations

IN THIS SECTION

Syntax

```
show security ipsec security-associations
<brief | detail>
<family (inet  | inet6)>
<fpc slot-number pic slot-number>
<index SA-index-number>
<kmd-instance (all | kmd-instance-name)>
<node-local>
<pic slot-number fpc slot-number>
<sa-type shortcut>
<traffic-selector traffic-selector-name>
<srg-id id-number>
<vpn-name vpn-name>
<ha-link-encryption>
```

Description

Display information about the IPsec security associations (SAs).

In Junos OS Releases 20.1R2, 20.2R2, 20.3R2, 20.3R1, and later, when you execute the `show security ipsec security-associations detail` command, a new output field `IKE SA Index` corresponding to every IPsec SA within a tunnel is displayed under each IPsec SA information. See "show security ipsec security-associations detail (SRX5400, SRX5600, SRX5800)" on page 1927.

Options

none

Display information about all SAs.

`brief | detail`

(Optional) Display the specified level of output. The default is `brief`.

`family`

(Optional) Display SAs by family. This option is used to filter the output.

- `inet`—IPv4 address family.
- `inet6`—IPv6 address family.

`fpc` ***`slot-number`*****`pic`** ***`slot-number`***

(Optional) Display information about existing IPsec SAs in the specified Flexible PIC Concentrator (FPC) slot and PIC slot.

In a chassis cluster, when you execute the CLI command `show security ipsec security-associations pic` *`<slot-number>`* `fpc` *`<slot-number>`* in operational mode, only the primary node information about the existing IPsec SAs in the specified Flexible PIC Concentrator (FPC) slot and PIC slot is displayed.

`index` ***`SA-index-number`***

(Optional) Display detailed information about the specified SA identified by this index number. To obtain a list of all SAs that includes their index numbers, use the command with no options.

`kmd-instance`

(Optional) Display information about existing IPsec SAs in the key management process (in this case, it is KMD) identified by the FPC *slot-number* and PIC *slot-number*.

- `all`—All KMD instances running on the Services Processing Unit (SPU).
- *`kmd-instance-name`*—Name of the KMD instance running on the SPU.

node-local —(Optional) Display information about IPsec SAs for node-local tunnels in a Multinode High Availability setup.

`pic` *`slot-number`*`fpc` *`slot-number`* (Optional) Display information about existing IPsec SAs in the specified PIC slot and FPC slot.

`sa-type` (Optional for ADVPN) Display information for the specified type of SA. `shortcut` is the only option for this release.

`traffic-selector` *`traffic-selector-name`* (Optional) Display information about the specified traffic selector.

`vpn-name` *`vpn-name`* (Optional) Display information about the specified VPN.

ha-link-encryption (Optional) Display information related to interchassis link tunnel only. See "ipsec (High Availability)" on page 1544, "show security ipsec security-associations ha-link-encryption (SRX5400, SRX5600, SRX5800)" on page 1928, and "show security ipsec sa detail ha-link-encryption (SRX5400, SRX5600, SRX5800)" on page 1929.

srg-id (Optional) Display information related to a specific services redundancy group (SRG) in a Multinode High Availability setup.

Required Privilege Level

view

Output Fields

Table 150 on page 1902 lists the output fields for the `show security ipsec security-associations` command, Table 151 on page 1907 lists the output fields for the `show security ipsec sa` command and Table 152 on page 1909. lists the output fields for the `show security ipsec sa detail`. Output fields are listed in the approximate order in which they appear.

Table 150: show security ipsec security-associations

Field Name	Field Description	Level of Output
`Total active tunnels`	Total number of active IPsec tunnels.	`brief`
`ID`	Index number of the SA. You can use this number to get additional information about the SA.	All levels
`Algorithm`	Cryptography used to secure exchanges between peers during the IKE negotiations includes: • An authentication algorithm used to authenticate exchanges between the peers. • An encryption algorithm used to encrypt data traffic.	`brief`
`SPI`	Security parameter index (SPI) identifier. An SA is uniquely identified by an SPI. Each entry includes the name of the VPN, the remote gateway address, the SPIs for each direction, the encryption and authentication algorithms, and keys. The peer gateways each have two SAs, one resulting from each of the two phases of negotiation: IKE and IPsec.	`brief`
`Life: sec/kb`	The lifetime of the SA, after which it expires, expressed either in seconds or kilobytes.	`brief`
`Mon`	The Mon field refers to VPN monitoring status. If VPN monitoring is enabled, then this field displays `U` (up) or `D` (down). A hyphen (-) means VPN monitoring is not enabled for this SA. A `V` means that IPsec datapath verification is in progress.	`brief`

Table 150: show security ipsec security-associations *(Continued)*

Field Name	Field Description	Level of Output
`lsys`	The root system.	`brief`
`Port`	If Network Address Translation (NAT) is used, this value is 4500. Otherwise, it is the standard IKE port, 500.	All levels
`Gateway`	IP address of the remote gateway.	`brief`
`Virtual-system`	Name of the logical system.	`detail`
`VPN name`	IPsec name for VPN.	`detail`
`State`	State has two options, `Installed` and `Not Installed`. • `Installed`—The SA is installed in the SA database. • `Not Installed`—The SA is not installed in the SA database. For transport mode, the value of State is always `Installed`.	`detail`
`Local gateway`	Gateway address of the local system.	`detail`
`Remote gateway`	Gateway address of the remote system.	`detail`
`Traffic selector`	Name of the traffic selector.	`detail`
`Local identity`	Identity of the local peer so that its partner destination gateway can communicate with it. The value is specified as an IP address, fully qualified domain name, e-mail address, or distinguished name (DN).	`detail`

Table 150: show security ipsec security-associations *(Continued)*

Field Name	Field Description	Level of Output
`Remote identity`	IP address of the destination peer gateway.	`detail`
`Term`	Defines local IP range, remote IP range, source port range, destination port range, and protocol.	`detail`
`Source-port`	Source port range configured for a term.	`detail`
`Destination-Port`	Destination port range configured for a term.	`detail`
`Version`	IKE version, either `IKEv1` or `IKEv2`.	`detail`
`DF-bit`	State of the don't fragment bit: `set` or `cleared`.	`detail`
`Location`	`FPC`—Flexible PIC Concentrator (FPC) slot number. `PIC`—PIC slot number. `KMD-Instance`—The name of the KMD instance running on the SPU, identified by FPC *slot-number* and PIC *slot-number*. Currently, 4 KMD instances running on each SPU, and any particular IPsec negotiation is carried out by a single KMD instance.	`detail`
`Tunnel events`	Tunnel event and the number of times the event has occurred. See Tunnel Events for descriptions of tunnel events and the action you can take.	`detail`
`Anchorship`	Anchor thread ID for the SA (for SRX4600 Series devices with the `detail` option).	

Table 150: show security ipsec security-associations *(Continued)*

Field Name	Field Description	Level of Output
`Direction`	Direction of the SA; it can be inbound or outbound.	`detail`
`AUX-SPI`	Value of the auxiliary security parameter index(SPI). • When the value is `AH` or `ESP`, `AUX-SPI` is always 0. • When the value is `AH+ESP`, `AUX-SPI` is always a positive integer.	`detail`
`Mode`	Mode of the SA: • `transport`—Protects host-to-host connections. • `tunnel`–Protects connections between security gateways.	`detail`
`Type`	Type of the SA: • `manual`—Security parameters require no negotiation. They are static and are configured by the user. • `dynamic`—Security parameters are negotiated by the IKE protocol. Dynamic SAs are not supported in transport mode.	`detail`
`State`	State of the SA: • `Installed`—The SA is installed in the SA database. • `Not Installed`—The SA is not installed in the SA database. For transport mode, the value of State is always `Installed`.	`detail`

Table 150: show security ipsec security-associations *(Continued)*

Field Name	Field Description	Level of Output
`Protocol`	Protocol supported. - Transport mode supports Encapsulation Security Protocol (ESP) and Authentication Header (AH). - Tunnel mode supports ESP and AH.	`detail`
`Authentication`	Type of authentication used.	`detail`
`Encryption`	Type of encryption used. Starting in Junos OS Release 19.4R2, when you configure `aes-128-gcm` or `aes-256-gcm` as an encryption algorithm at the `[edit security ipsec proposal proposal-name]` hierarchy level, the authentication algorithm field of the `show security ipsec security-associations detail` command displays the same configured encryption algorithm.	`detail`
`Soft lifetime`	The soft lifetime informs the IPsec key management system that the SA is about to expire. Each lifetime of an SA has two display options, hard and soft, one of which must be present for a dynamic SA. This allows the key management system to negotiate a new SA before the hard lifetime expires. - `Expires in seconds`—Number of seconds left until the SA expires.	`detail`
`Hard lifetime`	The hard lifetime specifies the lifetime of the SA. - `Expires in seconds`—Number of seconds left until the SA expires.	`detail`

Table 150: show security ipsec security-associations *(Continued)*

Field Name	Field Description	Level of Output
`Lifesize Remaining`	The lifesize remaining specifies the usage limits in kilobytes. If there is no lifesize specified, it shows unlimited. - `Expires in kilobytes`—Number of kilobytes left until the SA expires.	`detail`
`Anti-replay service`	State of the service that prevents packets from being replayed. It can be `Enabled` or `Disabled`.	`detail`
`Replay window size`	Size of the antireplay service window, which is 64 bits.	`detail`
`Bind-interface`	The tunnel interface to which the route-based VPN is bound.	`detail`
`Copy-Outer-DSCP`	Indicates if the system copies the outer DSCP value from the IP header to the inner IP header.	`detail`
`tunnel-establishment`	Indicates how the IKE is activated.	`detail`
`IKE SA index`	Indicates the list of parent IKE security associations.	`detail`

Table 151: show security ipsec sa Output Fields

Field Name	Field Description
`Total active tunnels`	Total number of active IPsec tunnels.

Table 151: show security ipsec sa Output Fields *(Continued)*

Field Name	Field Description
`ID`	Index number of the SA. You can use this number to get additional information about the SA.
`Algorithm`	Cryptography used to secure exchanges between peers during the IKE Phase 2 negotiations includes: - An authentication algorithm used to authenticate exchanges between the peers. Options are `hmac-md5-96`, `hmac-sha-256-128`, or `hmac-sha1-96`. - An encryption algorithm used to encrypt data traffic. Options are `3des-cbc`, `aes-128-cbc`, `aes-192-cbc`, `aes-256-cbc`, or `des-cbc`.
`SPI`	Security parameter index (SPI) identifier. An SA is uniquely identified by an SPI. Each entry includes the name of the VPN, the remote gateway address, the SPIs for each direction, the encryption and authentication algorithms, and keys. The peer gateways each have two SAs, one resulting from each of the two phases of negotiation: Phase 1 and Phase 2.
`Life:sec/kb`	The lifetime of the SA, after which it expires, expressed either in seconds or kilobytes.
`Mon`	The Mon field refers to VPN monitoring status. If VPN monitoring is enabled, then this field displays U (up) or D (down). A hyphen (-) means VPN monitoring is not enabled for this SA. A V means that IPSec datapath verification is in progress.
`lsys`	The root system.
`Port`	If Network Address Translation (NAT) is used, this value is 4500. Otherwise, it is the standard IKE port, 500.
`Gateway`	Gateway address of the system.

Table 152: show security ipsec sa detail Output Fields

Field Name	Field Description
`ID`	Index number of the SA. You can use this number to get additional information about the SA.
`Virtual-system`	The virtual system name.
`VPN Name`	IPSec name for VPN.
`Local Gateway`	Gateway address of the local system.
`Remote Gateway`	Gateway address of the remote system.
`Local Identity`	Identity of the local peer so that its partner destination gateway can communicate with it. The value is specified as an IP address, fully qualified domain name, e-mail address, or distinguished name (DN).
`Remote Identity`	IP address of the destination peer gateway.
`Version`	IKE version. For example, IKEv1, IKEv2.
`Passive Mode Tunneling`	IPsec tunneling of malformed packets. You can either enable or disable the option.
`DF-bit`	State of the don't fragment bit: `set` or `cleared`.
`Bind-interface`	The tunnel interface to which the route-based VPN is bound.
Tunnel Events	
`Direction`	Direction of the SA; it can be inbound or outbound.

Table 152: show security ipsec sa detail Output Fields *(Continued)*

Field Name	Field Description
AUX-SPI	Value of the auxiliary security parameter index(SPI). • When the value is AH or ESP, AUX-SPI is always 0. • When the value is AH+ESP, AUX-SPI is always a positive integer.
VPN Monitoring	If VPN monitoring is enabled, then the Mon field displays U (up) or D (down). A hyphen (-) means VPN monitoring is not enabled for this SA. A V means that IPsec datapath verification is in progress.
Hard lifetime	The hard lifetime specifies the lifetime of the SA. • Expires in seconds - Number of seconds left until the SA expires.
Lifesize Remaining	The lifesize remaining specifies the usage limits in kilobytes. If there is no lifesize specified, it shows unlimited.
Soft lifetime	The soft lifetime informs the IPsec key management system that the SA is about to expire. Each lifetime of an SA has two display options, hard and soft, one of which must be present for a dynamic SA. This allows the key management system to negotiate a new SA before the hard lifetime expires. • Expires in seconds - Number of seconds left until the SA expires.
Mode	Mode of the SA: • transport - Protects host-to-host connections. • tunnel - Protects connections between security gateways.
Type	Type of the SA: • manual - Security parameters require no negotiation. They are static and are configured by the user. • dynamic - Security parameters are negotiated by the IKE protocol. Dynamic SAs are not supported in transport mode.

Table 152: show security ipsec sa detail Output Fields *(Continued)*

Field Name	Field Description
`State`	State of the SA: • `Installed` - The SA is installed in the SA database. • `Not Installed` - The SA is not installed in the SA database. For transport mode, the value of State is always Installed.
`Protocol`	Protocol supported. • Transport mode supports Encapsulation Security Protocol (ESP) and Authentication Header (AH). • Tunnel mode supports ESP and AH. • `Authentication` - Type of authentication used. • `Encryption` - Type of encryption used.
`Anti-replay service`	State of the service that prevents packets from being replayed. It can be `Enabled` or `Disabled`.
`Replay window size`	Configured size of the antireplay service window. It can be 32 or 64 packets. If the replay window size is 0, the antireplay service is disabled. The antireplay window size protects the receiver against replay attacks by rejecting old or duplicate packets.
Interchassis Link Tunnel	
HA Link Encryption Mode	High availability mode supported. Displays `Multi-Node` when multi-node high availability feature is enabled.

Sample Output

For brevity, the show command outputs does not display all the values of the configuration. Only a subset of the configuration is displayed. Rest of the configuration on the system has been replaced with ellipses (...).

show security ipsec security-associations (IPv4)

```
user@host> show security ipsec security-associations
  Total active tunnels: 14743 Total Ipsec sas: 14743
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <511672 ESP:aes-cbc-128/sha1 0x071b8cd2      -   root 500   10.21.45.152
  >503327 ESP:aes-cbc-128/sha1 0x69d364dd 1584/ unlim - root 500 10.21.12.255
  <503327 ESP:aes-cbc-128/sha1 0x0a577f2d 1584/ unlim - root 500 10.21.12.255
  >512896 ESP:aes-cbc-128/sha1 0xd2f51c81 1669/ unlim - root 500 10.21.50.96
  <512896 ESP:aes-cbc-128/sha1 0x071b8d9e 1669/ unlim - root 500 10.21.50.96
  >513881 ESP:aes-cbc-128/sha1 0x95955834 1696/ unlim - root 500 10.21.54.57
  <513881 ESP:aes-cbc-128/sha1 0x0a57860c 1696/ unlim - root 500 10.21.54.57
  >505835 ESP:aes-cbc-128/sha1 0xf827b5c6 1598/ unlim - root 500 10.21.22.204
  <505835 ESP:aes-cbc-128/sha1 0x0f43bf3f 1598/ unlim - root 500 10.21.22.204
  >506531 ESP:aes-cbc-128/sha1 0x01694572 1602/ unlim - root 500 10.21.25.131
  <506531 ESP:aes-cbc-128/sha1 0x0a578143 1602/ unlim - root 500 10.21.25.131
  >512802 ESP:aes-cbc-128/sha1 0xdc292de4 1668/ unlim - root 500 10.21.50.1
  <512802 ESP:aes-cbc-128/sha1 0x0a578558 1668/ unlim - root 500 10.21.50.1
  >512413 ESP:aes-cbc-128/sha1 0xbe2c52d5 1660/ unlim - root 500 10.21.48.125
  <512413 ESP:aes-cbc-128/sha1 0x1129580c 1660/ unlim - root 500 10.21.48.125
  >505075 ESP:aes-cbc-128/sha1 0x2aae6647 1593/ unlim - root 500 10.21.19.213
  <505075 ESP:aes-cbc-128/sha1 0x02dc5c50 1593/ unlim - root 500 10.21.19.213
  >514055 ESP:aes-cbc-128/sha1 0x2b8adfcb 1704/ unlim - root 500 10.21.54.238
  <514055 ESP:aes-cbc-128/sha1 0x0f43c49a 1704/ unlim - root 500 10.21.54.238
  >508898 ESP:aes-cbc-128/sha1 0xbcced4d6 1619/ unlim - root 500 10.21.34.194
  <508898 ESP:aes-cbc-128/sha1 0x1492035a 1619/ unlim - root 500 10.21.34.194
  >505328 ESP:aes-cbc-128/sha1 0x2a8d2b36 1594/ unlim - root 500 10.21.20.208
  <505328 ESP:aes-cbc-128/sha1 0x14920107 1594/ unlim - root 500 10.21.20.208
  >500815 ESP:aes-cbc-128/sha1 0xdd86c89a 1573/ unlim - root 500 10.21.3.47
  <500815 ESP:aes-cbc-128/sha1 0x1129507f 1573/ unlim - root 500 10.21.3.47
  >503758 ESP:aes-cbc-128/sha1 0x64cc490e 1586/ unlim - root 500 10.21.14.172
  <503758 ESP:aes-cbc-128/sha1 0x14920001 1586/ unlim - root 500 10.21.14.172
  >504004 ESP:aes-cbc-128/sha1 0xde0b63ee 1587/ unlim - root 500 10.21.15.164
  <504004 ESP:aes-cbc-128/sha1 0x071b87d4 1587/ unlim - root 500 10.21.15.164
  >508816 ESP:aes-cbc-128/sha1 0x2703b7a5 1618/ unlim - root 500 10.21.34.112
  <508816 ESP:aes-cbc-128/sha1 0x071b8af6 1618/ unlim - root 500 10.21.34.112
  >511341 ESP:aes-cbc-128/sha1 0x828f3330 1644/ unlim - root 500 10.21.44.77
  <511341 ESP:aes-cbc-128/sha1 0x02dc6064 1644/ unlim - root 500 10.21.44.77
  >500456 ESP:aes-cbc-128/sha1 0xa6f1515d 1572/ unlim - root 500 10.21.1.200
  <500456 ESP:aes-cbc-128/sha1 0x1491fddb 1572/ unlim - root 500 10.21.1.200
  >512506 ESP:aes-cbc-128/sha1 0x4108f3a3 1662/ unlim - root 500 10.21.48.218
  <512506 ESP:aes-cbc-128/sha1 0x071b8d5d 1662/ unlim - root 500 10.21.48.218
  >504657 ESP:aes-cbc-128/sha1 0x27a6b8b3 1591/ unlim - root 500 10.21.18.41
  <504657 ESP:aes-cbc-128/sha1 0x112952fe 1591/ unlim - root 500 10.21.18.41
  >506755 ESP:aes-cbc-128/sha1 0xc0afcff0 1604/ unlim - root 500 10.21.26.100
  <506755 ESP:aes-cbc-128/sha1 0x149201f5 1604/ unlim - root 500 10.21.26.100
  >508023 ESP:aes-cbc-128/sha1 0xa1a90af8 1612/ unlim - root 500 10.21.31.87
  <508023 ESP:aes-cbc-128/sha1 0x02dc5e3b 1612/ unlim - root 500 10.21.31.87
  >509190 ESP:aes-cbc-128/sha1 0xee52074d 1621/ unlim - root 500 10.21.35.230
  <509190 ESP:aes-cbc-128/sha1 0x0f43c16e 1621/ unlim - root 500 10.21.35.230
  >505051 ESP:aes-cbc-128/sha1 0x24130b1c 1593/ unlim - root 500 10.21.19.188
  <505051 ESP:aes-cbc-128/sha1 0x149200d9 1593/ unlim - root 500 10.21.19.188
  >513214 ESP:aes-cbc-128/sha1 0x2c4752d1 1676/ unlim - root 500 10.21.51.158
  <513214 ESP:aes-cbc-128/sha1 0x071b8dd3 1676/ unlim - root 500 10.21.0.51.158
  >510808 ESP:aes-cbc-128/sha1 0x4acd94d3 1637/ unlim - root 500 10.21.42.56
  <510808 ESP:aes-cbc-128/sha1 0x071b8c42 1637/ unlim - root 500 10.21.42.56
```

show security ipsec security-associations (IPv6)

```
user@host> show security ipsec security-associations
Total active tunnels: 1
ID     Algorithm         SPI      Life:sec/kb  Mon  vsys Port  Gateway
131074 ESP:aes256/sha256 14caf1d9 3597/ unlim   -   root 500   2001:db8::1112
131074 ESP:aes256/sha256 9a4db486 3597/ unlim   -   root 500   2001:db8::1112
```

show security ipsec security-associations index 511672

```
user@host> show security ipsec security-associations index 511672
ID: 511672 Virtual-system: root, VPN Name: ipsec_vpn
  Local Gateway: 10.20.0.1, Remote Gateway: 10.21.45.152
  Traffic Selector Name: ts
  Local Identity: ipv4(10.191.151.0-10.191.151.255)
  Remote Identity: ipv4(10.40.151.0-10.40.151.255)
  Version: IKEv2
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0, Policy-name: IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Location: FPC 0, PIC 1, KMD-Instance 0
  Anchorship: Thread 10
  Direction: inbound, SPI: 0x835b8b42, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1639 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1257 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 0x071b8cd2, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1639 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1257 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

show security ipsec security-associations index 131073 detail

```
user@host> show security ipsec security-associations index 131073 detail
ID: 131073 Virtual-system: root, VPN Name: IPSEC_VPN1
  Local Gateway: 10.4.0.1, Remote Gateway: 10.5.0.1
  Local Identity: ipv4_subnet(any:0,[0..7]=10.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=10.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1
  Port: 500, Nego#: 18, Fail#: 0, Def-Del#: 0 Flag: 0x600a39
  Multi-sa, Configured SAs# 9, Negotiated SAs#: 9
  Tunnel events:
    Mon Apr 23 2018 22:20:54 -0700: IPSec SA negotiation successfully completed (1 times)
    Mon Apr 23 2018 22:20:54 -0700: IKE SA negotiation successfully completed (2 times)
    Mon Apr 23 2018 22:20:18 -0700: User cleared IKE SA from CLI, corresponding IPSec SAs
cleared (1 times)
    Mon Apr 23 2018 22:19:55 -0700: IPSec SA negotiation successfully completed (2 times)
    Mon Apr 23 2018 22:19:23 -0700: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Mon Apr 23 2018 22:19:23 -0700: Bind-interface's zone received. Information updated (1 times)
    Mon Apr 23 2018 22:19:23 -0700: External interface's zone received. Information updated (1
times)
  Direction: inbound, SPI: 2d8e710b, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1563 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Multi-sa FC Name: default
  Direction: outbound, SPI: 5f3a3239, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1563 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Multi-sa FC Name: default
  Direction: inbound, SPI: 5d227e19, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1551 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Multi-sa FC Name: best-effort
  Direction: outbound, SPI: 5490da, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1930 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1551 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes-256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
...
```

Starting with Junos OS Release 18.2R1, the CLI `show security ipsec security-associations index` *`index-number`* `detail` output displays all the child SA details including forwarding class name.

show security ipsec sa

```
user@host> show security ipsec sa
Total active tunnels: 2
ID Algorithm SPI Life:sec/kb Mon lsys Port Gateway
>67108885 ESP:aes-gcm-256/None fdef4dab 2918/ unlim - root 500 2001:db8:3000::2
>67108885 ESP:aes-gcm-256/None e785dadc 2918/ unlim - root 500 2001:db8:3000::2
>67108887 ESP:aes-gcm-256/None 34a787af 2971/ unlim - root 500 2001:db8:5000::2
>67108887 ESP:aes-gcm-256/None cf57007f 2971/ unlim - root 500 2001:db8:5000::2
```

show security ipsec sa detail

```
user@host> show security ipsec sa detail
ID: 500201 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 10.2.0.1, Remote Gateway: 10.2.0.2
  Local Identity: ipv4(10.0.0.0-255.255.255.255)
  Remote Identity: ipv4(10.0.0.0-255.255.255.255)
  Version: IKEv1
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Location: FPC 0, PIC 1, KMD-Instance 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0x0a25c960, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 91 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 44 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    tunnel-establishment: establish-tunnels-responder-only-no-rekey
  Direction: outbound, SPI: 0x43e34ad3, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 91 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 44 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    tunnel-establishment: establish-tunnels-responder-only-no-rekey
...
```

Starting with Junos OS Release 19.1R1, a new field **tunnel-establishment** in the output of the CLI `show security ipsec sa detail` displays the option configured under `ipsec vpn establish-tunnels` hierarchy.

Starting with Junos OS Release 21.3R1, a new field **Tunnel MTU** in the output of the CLI `show security ipsec sa detail` displays the option configured under `ipsec vpn hub-to-spoke-vpn tunnel-mtu` hierarchy.

Starting in Junos OS Release 22.1R3, on SRX5000 line, the Tunnel MTU is not displayed in the CLI output if the tunnel MTU is not configured.

show security ipsec sa details (MX-SPC3)

```
user@host> show security ipsec sa detail
ID: 500055 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 10.2.0.1, Remote Gateway: 10.2.0.2
  Local Identity: ipv4(10.0.0.0-255.255.255.255)
  Remote Identity: ipv4(10.0.0.0-255.255.255.255)
  Version: IKEv2
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Tunnel MTU: 1420  Policy-name:
IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Location: FPC 0, PIC 0, KMD-Instance 0
  Anchorship: Thread 15
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0x229b998e, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 23904 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 23288 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-md5-96, Encryption: aes-cbc (128 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Enabled
    tunnel-establishment: establish-tunnels-immediately
  Direction: outbound, SPI: 0xb2e843a3, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 23904 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 23288 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-md5-96, Encryption: aes-cbc (128 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Enabled
    tunnel-establishment: establish-tunnels-immediately
```

show security ipsec sa details (MX-SPC3) with passive mode tunneling

```
user@host> show security ipsec sa detail
ID: 500054 Virtual-system: root, VPN Name: TUN_3
  Local Gateway: 100.0.0.3, Remote Gateway: 200.0.0.3
  Traffic Selector Name: ts1
  Local Identity: ipv4(11.0.0.3-11.0.0.3)
  Remote Identity: ipv4(75.0.0.3-75.0.0.3)
  TS Type: traffic-selector
  Version: IKEv2
  Quantum Secured: No
  PFS group: N/A
  SRG ID: 0
  Passive mode tunneling: Enabled
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.3, Policy-name: IPSEC_POLICY
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Tunnel events:
    Mon Sep 19 2022 19:27:44: IPsec SA negotiation succeeds (1 times)
  Location: FPC 3, PIC 1, KMD-Instance 0
  Anchorship: Thread 15
  Distribution-Profile: vms-3/1/0
  Direction: inbound, SPI: 0x25c03740, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expired
    Lifesize Remaining: Expired
    Soft lifetime: Expires in 2920 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 512
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    IKE SA Index: 122
  Direction: outbound, SPI: 0x8e8f2009, AUX-SPI: 0
```

```
                              , VPN Monitoring: -
    Hard lifetime: Expired
    Lifesize Remaining: Expired
    Soft lifetime: Expires in 2920 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 512
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    IKE SA Index: 122
```

show security ipsec security-association

```
user@host>show security ipsec security-association
Total active tunnels: 1     Total IPsec sas: 1
  ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <500006 ESP:aes-gcm-128/aes128-gcm 0x782b233c 1432/ unlim - root 500 10.2.0.2
```

show security ipsec security-associations brief

```
user@host> show security ipsec security-associations brief
Total active tunnels: 2     Total Ipsec sas: 18
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131073 ESP:aes256/sha256 89e5098  1569/ unlim   -   root 500   10.5.0.1
  >131073 ESP:aes256/sha256 fcee9d54 1569/ unlim   -   root 500   10.5.0.1
  <131073 ESP:aes256/sha256 f3117676 1609/ unlim   -   root 500   10.5.0.1
  >131073 ESP:aes256/sha256 6050109f 1609/ unlim   -   root 500   10.5.0.1
  <131073 ESP:aes256/sha256 e01f54b1 1613/ unlim   -   root 500   10.5.0.1
  >131073 ESP:aes256/sha256 29a05dd6 1613/ unlim   -   root 500   10.5.0.1
  <131073 ESP:aes256/sha256 606c90f6 1616/ unlim   -   root 500   10.5.0.1
  >131073 ESP:aes256/sha256 9b5b059d 1616/ unlim   -   root 500   10.5.0.1
  <131073 ESP:aes256/sha256 b8116d6d 1619/ unlim   -   root 500   10.5.0.1
  >131073 ESP:aes256/sha256 b7ed6bfd 1619/ unlim   -   root 500   10.5.0.1
  <131073 ESP:aes256/sha256 4f5ce754 1619/ unlim   -   root 500   10.5.0.1
  >131073 ESP:aes256/sha256 af8984b6 1619/ unlim   -   root 500   10.5.0.1
...
```

show security ipsec security-associations detail

```
user@host> show security ipsec security-associations detail

ID: 500009 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 10.2.0.2, Remote Gateway: 10.2.0.1
  Local Identity: ipv4(10.0.0.0-255.255.255.255)
  Remote Identity: ipv4(10.0.0.0-255.255.255.255)
  Version: IKEv1
  PFS group: DH-group-14
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Location: FPC 0, PIC 0, KMD-Instance 0
  Anchorship: Thread 0
  Distribution-Profile: default-profile
  IKE SA Index: 2068
  Direction: inbound, SPI: 0xba7bb1f2, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 146 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 101 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-on-traffic
  Direction: outbound, SPI: 0x41650a1b, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 146 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 101 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-on-traffic
```

show security ipsec security-associations family inet6

```
user@host> show security ipsec security-associations family inet6
  Virtual-system: root
  Local Gateway: 2001:db8:1212::1111, Remote Gateway: 2001:db8:1212::1112
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    DF-bit: clear
    Direction: inbound, SPI: 14caf1d9, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3440 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2813 seconds
    Mode: tunnel, Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64

    Direction: outbound, SPI: 9a4db486, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3440 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2813 seconds
    Mode: tunnel, Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc
    Anti-replay service: counter-based enabled, Replay window size: 64
```

show security ipsec security-associations fpc 6 pic 1 kmd-instance all (SRX Series Firewalls)

```
user@host> show security ipsec security-associations fpc 6 pic 1 kmd-instance all
  Total active tunnels: 1

ID    Gateway          Port  Algorithm           SPI      Life:sec/kb  Mon vsys

<2    192.168.1.2      500   ESP:aes256/sha256   67a7d25d 28280/unlim   -   0

>2    192.168.1.2      500   ESP:aes256/sha256   a23cbcdc 28280/unlim   -   0
```

show security ipsec security-associations detail (ADVPN Suggester, Static Tunnel)

```
user@host> show security ipsec security-associations detail
ID: 70516737 Virtual-system: root, VPN Name: ZTH_HUB_VPN
  Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear
  Bind-interface: st0.1

  Port: 500, Nego#: 5, Fail#: 0, Def-Del#: 0 Flag: 0x608a29
  Tunnel events:
  Tue Nov 03 2015 01:24:27 -0800: IPSec SA negotiation successfully completed (1 times)
  Tue Nov 03 2015 01:24:27 -0800: IKE SA negotiation successfully completed (4 times)
  Tue Nov 03 2015 01:23:38 -0800: User cleared IPSec SA from CLI (1 times)
  Tue Nov 03 2015 01:21:32 -0800: IPSec SA negotiation successfully completed (1 times)
  Tue Nov 03 2015 01:21:31 -0800: IPSec SA delete payload received from peer, corresponding
IPSec SAs cleared (1 times)
  Tue Nov 03 2015 01:21:27 -0800: IPSec SA negotiation successfully completed (1 times)
  Tue Nov 03 2015 01:21:13 -0800: Tunnel configuration changed. Corresponding IKE/IPSec SAs are
deleted (1 times)
  Tue Nov 03 2015 01:19:27 -0800: IPSec SA negotiation successfully completed (1 times)
  Tue Nov 03 2015 01:19:27 -0800: Tunnel is ready. Waiting for trigger event or peer to trigger
negotiation (1 times)
  Location: FPC 0, PIC 3, KMD-Instance 2
  Direction: inbound, SPI: 43de5d65, AUX-SPI: 0
  Hard lifetime: Expires in 1335 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 996 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc (256 bits)
  Anti-replay service: counter-based enabled

  , Replay window size: 64
  Location: FPC 0, PIC 3, KMD-Instance 2
  Direction: outbound, SPI: 5b6e157c, AUX-SPI: 0
  Hard lifetime: Expires in 1335 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 996 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc (256 bits)
  Anti-replay service: counter-based enabled


  , Replay window size: 64
```

show security ipsec security-associations detail (ADVPN Partner, Static Tunnel)

```
user@host> show security ipsec security-associations detail
ID: 67108872 Virtual-system: root, VPN Name: ZTH_SPOKE_VPN
  Local Gateway: 192.168.1.2, Remote Gateway: 192.168.1.1
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x8608a29
  Tunnel events:
  Tue Nov 03 2015 01:24:26 -0800: IPSec SA negotiation successfully completed (1 times)
  Tue Nov 03 2015 01:24:26 -0800: IKE SA negotiation successfully completed (4 times)
  Tue Nov 03 2015 01:23:37 -0800: IPSec SA delete payload received from peer, corresponding
IPSec SAs cleared (1 times)
  Tue Nov 03 2015 01:21:31 -0800: IPSec SA negotiation successfully completed (1 times)
  Tue Nov 03 2015 01:21:31 -0800: Tunnel is ready. Waiting for trigger event or peer to trigger
negotiation (1 times)
  Tue Nov 03 2015 01:18:26 -0800: Key pair not found for configured local certificate.
Negotiation failed (1 times)
  Tue Nov 03 2015 01:18:13 -0800: CA certificate for configured local certificate not found.
Negotiation not initiated/successful (1 times)
  Direction: inbound, SPI: 5b6e157c, AUX-SPI: 0
  Hard lifetime: Expires in 941 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 556 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: 43de5d65, AUX-SPI: 0
  Hard lifetime: Expires in 941 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 556 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64
```

show security ipsec security-associations sa-type shortcut (ADVPN)

```
user@host> show security ipsec security-associations sa-type shortcut
Total active tunnels: 1
ID         Algorithm        SPI      Life:sec/kb  Mon lsys Port  Gateway
<268173318 ESP:aes256/sha256 6f164ee0 3580/ unlim - root 500 192.168.0.111
>268173318 ESP:aes256/sha256 e6f29cb0 3580/ unlim - root 500 192.168.0.111
```

show security ipsec security-associations sa-type shortcut detail (ADVPN)

```
user@host> show security ipsec security-associations sa-type shortcut detail
node0:
--------------------------------------------------------------------------


ID: 67108874 Virtual-system: root, VPN Name: ZTH_SPOKE_VPN
  Local Gateway: 192.168.1.2, Remote Gateway: 192.168.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Auto Discovery VPN:
    Type: Shortcut, Shortcut Role: Initiator
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.1
  Port: 4500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0x40608a29
  Tunnel events:
    Tue Nov 03 2015 01:47:26 -0800: IPSec SA negotiation successfully completed (1 times)
    Tue Nov 03 2015 01:47:26 -0800: Tunnel is ready. Waiting for trigger event or peer to
trigger negotiation (1 times)
    Tue Nov 03 2015 01:47:26 -0800: IKE SA negotiation successfully completed (1 times)
  Direction: inbound, SPI: b7a5518, AUX-SPI: 0
    Hard lifetime: Expires in 1766 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1381 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Direction: outbound, SPI: b7e0268, AUX-SPI: 0
    Hard lifetime: Expires in 1766 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1381 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

show security ipsec security-associations family inet detail

```
user@host> show security ipsec security-associations family inet detail
ID: 131073 Virtual-system: root, VPN Name: ike-vpn
 Local Gateway: 192.168.1.1, Remote Gateway: 192.168.1.2
 Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
 Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
 Version: IKEv1
 DF-bit: clear
 , Copy-Outer-DSCP Enabled
 Bind-interface: st0.99

 Port: 500, Nego#: 116, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
 Tunnel events:
 Fri Oct 30 2015 15:47:21 -0700: IPSec SA rekey successfully completed (115 times)
 Fri Oct 30 2015 11:38:35 -0700: IKE SA negotiation successfully completed (12 times)
 Mon Oct 26 2015 16:41:07 -0700: IPSec SA negotiation successfully completed (1 times)
 Mon Oct 26 2015 16:40:56 -0700: Tunnel is ready. Waiting for trigger event or peer to trigger
negotiation (1 times)
 Mon Oct 26 2015 16:40:56 -0700: External interface's address received. Information updated (1
times)
 Location: FPC 0, PIC 1, KMD-Instance 1
 Direction: inbound, SPI: 81b9fc17, AUX-SPI: 0
 Hard lifetime: Expires in 1713 seconds
 Lifesize Remaining:  Unlimited
 Soft lifetime: Expires in 1090 seconds
 Mode: Tunnel(0 0), Type: dynamic, State: installed
 Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc (256 bits)
 Anti-replay service: counter-based enabled

 , Replay window size: 64
 Location: FPC 0, PIC 1, KMD-Instance 1
 Direction: outbound, SPI: 727f629d, AUX-SPI: 0
 Hard lifetime: Expires in 1713 seconds
 Lifesize Remaining:  Unlimited
 Soft lifetime: Expires in 1090 seconds
 Mode: Tunnel(0 0), Type: dynamic, State: installed
 Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc (256 bits)
```

```
  Anti-replay service: counter-based enabled


  , Replay window size: 64
```

show security ipsec security-associations detail (SRX4600)

```
user@host> show security ipsec security-associations detail
ID: 131073 Virtual-system: root, VPN Name: ike-vpn
  Local Gateway: 10.62.1.3, Remote Gateway: 10.62.1.2
  Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Version: IKEv2
  DF-bit: clear, Bind-interface: st0.0
  Port: 500, Nego#: 25, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
  Tunnel events:
    Fri Jan 12 2007 07:50:10 -0800: IPSec SA rekey successfully completed (23 times)
  Location: FPC 0, PIC 0, KMD-Instance 0
  Anchorship: Thread 6
  Direction: inbound, SPI: 812c9c01, AUX-SPI: 0
    Hard lifetime: Expires in 2224 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1598 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
  Location: FPC 0, PIC 0, KMD-Instance 0
  Anchorship: Thread 7
  Direction: outbound, SPI: c4de0972, AUX-SPI: 0
    Hard lifetime: Expires in 2224 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1598 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha-256, Encryption: aes256-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
```

show security ipsec security-associations detail (SRX5400, SRX5600, SRX5800)

A new output field `IKE SA Index` corresponding to every IPsec SA within a tunnel is displayed under each IPsec SA information.

```
user@host> show security ipsec security-associations detail
ID: 500005 Virtual-system: root, VPN Name: 85BX5-OAM
  Local Gateway: 10.217.0.4, Remote Gateway: 10.200.254.118
  Traffic Selector Name: TS_DEFAULT
  Local Identity: ipv4(0.0.0.0-255.255.255.255)
  Remote Identity: ipv4(10.181.235.224-10.181.235.224)
  Version: IKEv2
  PFS group: N/A
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0, Policy-name: MACRO-IPSEC-POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Location: FPC 7, PIC 1, KMD-Instance 0
  Anchorship: Thread 15
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0xe2eb3838, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 644 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 159 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes128-gcm, Encryption: aes-gcm (128 bits)
    Anti-replay service: disabled
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
    IKE SA Index: 22
  Direction: outbound, SPI: 0x4f7c3101, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 644 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 159 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes128-gcm, Encryption: aes-gcm (128 bits)
    Anti-replay service: disabled
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
    IKE SA Index: 22
  Direction: inbound, SPI: 0x30b6d66f, AUX-SPI: 0
```

```
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1771 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1391 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes128-gcm, Encryption: aes-gcm (128 bits)
    Anti-replay service: disabled
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
          IKE SA Index: 40
Direction: outbound, SPI: 0xd2db4108, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1771 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1391 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes128-gcm, Encryption: aes-gcm (128 bits)
    Anti-replay service: disabled
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-responder-only
          IKE SA Index: 40
```

show security ipsec security-associations ha-link-encryption (SRX5400, SRX5600, SRX5800)

Starting in Junos OS Release 20.4R1, when you configure the high availability (HA) feature, you can use this show command to view only interchassis link tunnel details.

```
user@host> show security ipsec security-associations ha-link-encryption
  Total active tunnels: 1     Total IPsec sas: 91
  ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <495001 ESP:aes-gcm-256/aes256-gcm 0x0047658d 298/ unlim - root 500 10.23.0.2
  >495001 ESP:aes-gcm-256/aes256-gcm 0x0046c5cd 298/ unlim - root 500 10.23.0.2
  <495001 ESP:aes-gcm-256/aes256-gcm 0x0447658d 298/ unlim - root 500 10.23.0.2
  >495001 ESP:aes-gcm-256/aes256-gcm 0x0446c5cd 298/ unlim - root 500 10.23.0.2
  <495001 ESP:aes-gcm-256/aes256-gcm 0x0847658d 298/ unlim - root 500 10.23.0.2
  >495001 ESP:aes-gcm-256/aes256-gcm 0x0846c5cd 298/ unlim - root 500 10.23.0.2
  <495001 ESP:aes-gcm-256/aes256-gcm 0x0c47658d 298/ unlim - root 500 10.23.0.2
  >495001 ESP:aes-gcm-256/aes256-gcm 0x0c46c5cd 298/ unlim - root 500 10.23.0.2
  <495001 ESP:aes-gcm-256/aes256-gcm 0x1047658d 298/ unlim - root 500 10.23.0.2
  >495001 ESP:aes-gcm-256/aes256-gcm 0x1046c5cd 298/ unlim - root 500 10.23.0.2
  <495001 ESP:aes-gcm-256/aes256-gcm 0x1447658d 298/ unlim - root 500 10.23.0.2
  >495001 ESP:aes-gcm-256/aes256-gcm 0x1446c5cd 298/ unlim - root 500 10.23.0.2
  <495001 ESP:aes-gcm-256/aes256-gcm 0x1847658d 298/ unlim - root 500 10.23.0.2
  >495001 ESP:aes-gcm-256/aes256-gcm 0x1846c5cd 298/ unlim - root 500 10.23.0.2
  <495001 ESP:aes-gcm-256/aes256-gcm 0x1c47658d 298/ unlim - root 500 10.23.0.2
  >495001 ESP:aes-gcm-256/aes256-gcm 0x1c46c5cd 298/ unlim - root 500 10.23.0.2
  <495001 ESP:aes-gcm-256/aes256-gcm 0x2047658d 298/ unlim - root 500 10.23.0.2
  >495001 ESP:aes-gcm-256/aes256-gcm 0x2046c5cd 298/ unlim - root 500 10.23.0.2
  <495001 ESP:aes-gcm-256/aes256-gcm 0x2447658d 298/ unlim - root 500 10.23.0.2
  >495001 ESP:aes-gcm-256/aes256-gcm 0x2446c5cd 298/ unlim - root 500 10.23.0.2
...
```

show security ipsec sa detail ha-link-encryption (SRX5400, SRX5600, SRX5800)

Starting in Junos OS Release 20.4R1, when you configure the high availability (HA) feature, you can use this show command to view only interchassis link tunnel details. It displays the multi SAs created for interchassis link encryption tunnel.

```
user@host> show security ipsec sa detail ha-link-encryption
ID: 495001 Virtual-system: root, VPN Name: L3HA_IPSEC_VPN
  Local Gateway: 10.23.0.1, Remote Gateway: 10.23.0.2
  Traffic Selector Name: __L3HA_IPSEC_VPN__multi_node__
  Local Identity: ipv4(180.100.1.1-180.100.1.1)
  Remote Identity: ipv4(180.100.1.2-180.100.1.2)
  Version: IKEv2
  PFS group: DH-Group-24
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.16000, Policy-name: L3HA_IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  HA Link Encryption Mode: Multi-Node
  Location: FPC -, PIC -, KMD-Instance -
  Anchorship: Thread -
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0x00439cf8, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 294 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 219 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    Location: FPC 1, PIC 0, KMD-Instance 0
    Anchorship: Thread 15
    IKE SA Index: 4294966297
  Direction: outbound, SPI: 0x004cfceb, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 294 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 219 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    Location: FPC 1, PIC 0, KMD-Instance 0
    Anchorship: Thread 15
    IKE SA Index: 4294966297
  Direction: inbound, SPI: 0x04439cf8, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 294 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 219 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    Location: FPC 1, PIC 0, KMD-Instance 0
    Anchorship: Thread 16
    IKE SA Index: 4294966297
  Direction: outbound, SPI: 0x044cfceb, AUX-SPI: 0
                              , VPN Monitoring: -

...
```

In Junos OS Release 22.3R1 and later, when you configure the Chassis Cluster HA control link encryption feature, you can execute the `show security ike sa ha-link-encryption detail`, `show security ipsec sa ha-link-encryption detail`, and `show security ipsec sa ha-link-encryption` commands to view the Chassis cluster control link encryption tunnel details.

show security ike sa ha-link-encryption detail

```
user@host> show security ike sa ha-link-encryption detail
IKE peer 10.2.0.1, Index 4294966274, Gateway Name: IKE_GW_HA_0
  Role: Initiator, State: UP
  Initiator cookie: ae5bcb5540d388a1, Responder cookie: 28bbae629ceb727f
  Exchange type: IKEv2, Authentication method: Pre-shared-keys
  Local gateway interface: em0
  Routing instance: __juniper_private1__
  Local: 10.7.0.2:500, Remote: 10.2.0.1:500
  Lifetime: Expires in 24856 seconds
  Reauth Lifetime: Disabled
  IKE Fragmentation: Enabled, Size: 576
  Remote Access Client Info: Unknown Client
  Peer ike-id: 10.2.0.1
  AAA assigned IP: 0.0.0.0
  Algorithms:
   Authentication        : hmac-sha1-96
   Encryption            : aes256-cbc
   Pseudo random function: hmac-sha1
   Diffie-Hellman group  : DH-group-2
  Traffic statistics:
   Input  bytes  :               200644
   Output bytes  :               200644
   Input  packets:                 2635
   Output packets:                 2635
   Input  fragmented packets:       0
   Output fragmented packets:       0
  IPSec security associations: 6 created, 3 deleted
  Phase 2 negotiations in progress: 1
  IPSec Tunnel IDs: 495002
    Negotiation type: Quick mode, Role: Initiator, Message ID: 0
    Local: 10.7.0.2:500, Remote: 10.2.0.1:500
    Local identity: 10.7.0.2
    Remote identity: 10.2.0.1
    Flags: IKE SA is created
 IPsec SA Rekey CREATE_CHILD_SA exchange stats:
   Initiator stats:                                  Responder stats:
    Request Out             : 1                       Request In             : 1
    Response In             : 1                       Response Out           : 1
    No Proposal Chosen In   : 0                       No Proposal Chosen Out : 0
    Invalid KE In           : 0                       Invalid KE Out         : 0
    TS Unacceptable In      : 0                         TS Unacceptable Out    : 0
    Res DH Compute Key Fail : 0                         Res DH Compute Key Fail: 0
    Res Verify SA Fail      : 0
    Res Verify DH Group Fail: 0
    Res Verify TS Fail      : 0
```

show security ipsec sa ha-link-encryption detail

```
user@host> show security ipsec sa ha-link-encryption detail
ID: 495002 Virtual-system: root, VPN Name: IPSEC_VPN_HA_0
  Local Gateway: 10.7.0.2, Remote Gateway: 10.2.0.1
  Traffic Selector Name: __IPSEC_VPN_HA_0__l2_chassis_clu
  Local Identity: ipv4(10.7.0.2-10.7.0.2)
  Remote Identity: ipv4(10.2.0.1-10.2.0.1)
  TS Type: traffic-selector
  Version: IKEv2
  PFS group: DH-group-24
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.16000, Tunnel MTU: 0, Policy-
name: IPSEC_POL_HA_0
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  HA Link Encryption Mode: L2 Chassis Cluster
  Location: FPC -, PIC -, KMD-Instance -
  Anchorship: Thread -
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0x35fae26b, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3435 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2818 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    IKE SA Index: 4294966274
  Direction: outbound, SPI: 0x0a2b9927, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 3435 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 2818 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    IKE SA Index: 4294966274
```

show security ipsec sa ha-link-encryption

```
user@host> show security ipsec sa ha-link-encryption
Total active tunnels: 1     Total IPsec sas: 1
  ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <495002 ESP:aes-cbc-256/sha1 0x35fae26b 3484/ unlim - root 500 10.2.0.1
  >495002 ESP:aes-cbc-256/sha1 0x0a2b9927 3484/ unlim - root 500 10.2.0.1
```

show security ipsec security-associations detail (SRX Series Firewalls and MX Series Routers)

In Junos OS Release 20.4R2, 21.1R1, and later, you can execute the `show security ipsec security-associations detail` command to view the traffic selector type for a VPN.

```
user@host> show security ipsec security-associations detail
ID: 500024 Virtual-system: root, VPN Name: S2S_VPN2
  Local Gateway: 10.7.0.2, Remote Gateway: 10.2.0.1
  Traffic Selector Name: ts1
  Local Identity: ipv4(10.20.20.0-10.20.20.255)
  Remote Identity: ipv4(10.10.10.0-10.10.10.255)
  TS Type: traffic-selector
  Version: IKEv2
  PFS group: DH-group-14
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.2, Policy-name: IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Tunnel events:
    Tue Jan 19 2021 04:43:49: IPsec SA negotiation succeeds (1 times)
  Location: FPC 0, PIC 0, KMD-Instance 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0xf8642fae, AUX-SPI: 0
```

```
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1798 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1397 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    IKE SA Index: 17
  Direction: outbound, SPI: 0xb2a26969, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1798 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1397 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    IKE SA Index: 17
ID: 500025 Virtual-system: root, VPN Name: S2S_VPN1
  Local Gateway: 10.7.0.1, Remote Gateway: 10.2.0.1
  Local Identity: ipv4(0.0.0.0-255.255.255.255)
  Remote Identity: ipv4(0.0.0.0-255.255.255.255)
  TS Type: proxy-id
  Version: IKEv2
  PFS group: DH-group-14
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  Tunnel events:
    Tue Jan 19 2021 04:44:41: IPsec SA negotiation succeeds (1 times)
  Location: FPC 0, PIC 0, KMD-Instance 0
  Anchorship: Thread 1
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0xe293762a, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1755 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1339 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    IKE SA Index: 18
  Direction: outbound, SPI: 0x7aef9d7f, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 1755 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 1339 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    IKE SA Index: 18
```

show security ipsec security-associations detail (SRX5400, SRX5600, SRX5800)

Starting in Junos OS Release 21.1R1, you can view the traffic selector details, that includes, local identity, remote identity, protocol, source-port range, destination port range for multiple terms defined for an IPsec SA.

In the earlier Junos Releases, traffic selection for a particular SA is performed using existing IP range defined using IP address or netmask. From Junos OS Release 21.1R1 onwards, additionally traffic is selected through protocol specified using *protocol_name*. And also, low and high port range specified for source and destination port numbers.

```
user@host> show security ipsec security-associations detail

ID: 500075 Virtual-system: root, VPN Name: pkn-r0-r1-ipsec-vpn-1
Local Gateway: 10.1.1.1, Remote Gateway: 10.1.1.2

Traffic Selector Name: ts1

   Local Identity:
   Protocol         Port                  IP
   17/UDP         100-200            198.51.100.0-198.51.100.255
   6/TCP         250-300            198.51.100.0-198.51.100.255
   Remote Identity:
   Protocol         Port                  IP
   17/UDP         150-200            10.80.0.1-10.80.0.1
   6/TCP         250-300            10.80.1.1-10.80.1.1
```

```
Version: IKEv2
DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0, Policy-name: pkn-r0-r1-ipsec-
policy
Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
Location: FPC 0, PIC 0, KMD-Instance 0
Anchorship: Thread 1
Distribution-Profile: default-profile
Direction: inbound, SPI: ………
Direction: outbound, SPI: …………
```

show security ipsec security-associations srg-id

```
user@host> show security ipsec security-associations srg-id 1

Total active tunnels: 1     Total IPsec sas: 2
  ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <17277217 ESP:aes-cbc-256/sha256 0xc7faee3e 1440/ unlim - root 500 10.112.0.1
  >17277217 ESP:aes-cbc-256/sha256 0x7921d472 1440/ unlim - root 500 10.112.0.1
  <17277217 ESP:aes-cbc-256/sha256 0xf1a01dd4 1498/ unlim - root 500 10.112.0.1
  >17277217 ESP:aes-cbc-256/sha256 0xa0b77273 1498/ unlim - root 500 10.112.0.1
```

show security ipsec security-associations node-local

```
user@host> show security ipsec security-associations node-local
Total active tunnels: 1     Total IPsec sas: 1
  ID      Algorithm              SPI      Life:sec/kb  Mon lsys Port   Gateway
  <500001 ESP:aes-cbc-256/sha256 0x5f2fdf60 3093/ unlim -  root 500   6.0.0.2
  >500001 ESP:aes-cbc-256/sha256 0x293e67e0 3093/ unlim -  root 500   6.0.0.2
```

show security ipsec security-associations node-local detail

```
user@host> show security ipsec security-associations node-local
ID: 500003 Virtual-system: root, VPN Name: IPSEC_VPN
  Local Gateway: 4.0.0.1, Remote Gateway: 6.0.0.2
  Local Identity: ipv4(0.0.0.0-255.255.255.255)
  Remote Identity: ipv4(0.0.0.0-255.255.255.255)
```

```
TS Type: proxy-id
Version: IKEv2
Quantum Secured: No
PFS group: DH-group-19
Passive mode tunneling: Disabled
DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.1, Policy-name: IPSEC_POL
Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
Tunnel events:
  Sun Apr 09 2023 22:22:27: IPsec SA negotiation succeeds (1 times)
Location: FPC 1, PIC 1, KMD-Instance 0
Anchorship: Thread 1
Distribution-Profile: default-profile
Direction: inbound, SPI: 0x8c8c3761, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3564 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 2884 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64
  Extended-Sequence-Number: Disabled
  tunnel-establishment: establish-tunnels-responder-only
  IKE SA Index: 25
Direction: outbound, SPI: 0x0e798f8c, AUX-SPI: 0
                            , VPN Monitoring: -
  Hard lifetime: Expires in 3564 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 2884 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
  Protocol: ESP, Authentication: hmac-sha256-128, Encryption: aes-cbc (256 bits)
  Anti-replay service: counter-based enabled, Replay window size: 64
  Extended-Sequence-Number: Disabled
  tunnel-establishment: establish-tunnels-responder-only
  IKE SA Index: 25
```

Release Information

Command introduced in Junos OS Release 8.5. Support for the `family` option added in Junos OS Release 11.1.

Support for the `vpn-name` option added in Junos OS Release 11.4R3. Support for the `traffic-selector` option and traffic selector field added in Junos OS Release 12.1X46-D10.

Support for Auto Discovery VPN (ADVPN) added in Junos OS Release 12.3X48-D10.

Support for IPsec datapath verification added in Junos OS Release 15.1X49-D70.

Support for thread anchorship added in Junos OS Release 17.4R1.

Starting in Junos OS Release 18.2R2 the `show security ipsec security-assocations detail` command output will include thread anchorship information for the security associations (SAs).

Starting in Junos OS Release 19.4R1, we have deprecated the CLI option `fc-name` (COS Forward Class name) in the new **iked** process that displays the security associations (SAs) under show command `show security ipsec sa`.

Support for the `ha-link-encryption` option added in Junos OS Release 20.4R1.

Support for the `srg-id` option added in Junos OS Release 22.4R1.

Support for the `passive-mode-tunneling` on MX-SPC3 is introduced in Junos OS Release 23.1R1.

Support for the `node-local` option is added in Junos OS Release 23.2R1.

RELATED DOCUMENTATION

Example: Configuring a Route-Based VPN Tunnel in a User Logical Systems

show security ipsec statistics

IN THIS SECTION

Syntax

```
show security ipsec statistics
<fpc slot-number>
<index SA-index-number>
<pic slot-number>
<srg-id id-number>
<ha-link-encryption>
```

Description

Display standard IPsec statistics.

Options

- none—Display statistics about all IPsec security associations (SAs).

- `fpc` *slot-number* —Specific to SRX Series Firewalls. Display statistics about existing IPsec SAs in this Flexible PIC Concentrator (FPC) slot. This option is used to filter the output.

- `index` *SA-index-number* —(Optional) Display statistics for the SA with this index number.

- `srg-id` *id-number* —(Optional) Display information related to a specific services redundancy group (SRG) in a Multinode High Availability setup.

- `pic` *`slot-number`* —Specific to SRX Series Firewalls. Display statistics about existing IPsec SAs in this PIC slot. This option is used to filter the output.
- `ha-link-encryption`—(Optional) Display information related to interchassis link tunnel only. See "ipsec (High Availability)" on page 1544 and "show security ipsec statistics ha-link-encryption (SRX5400, SRX5600, SRX5800)" on page 1944.

Required Privilege Level

view

Output Fields

Table 153 on page 1940 lists the output fields for the `show security ipsec statistics` command. Output fields are listed in the approximate order in which they appear.

Table 153: show security ipsec statistics Output Fields

Field Name	Field Description
`Virtual-system`	The root system.
`ESP Statistics`	• `Encrypted bytes`—Total number of bytes encrypted by the local system across the IPsec tunnel. • `Decrypted bytes`—Total number of bytes decrypted by the local system across the IPsec tunnel. • `Encrypted packets`—Total number of packets encrypted by the local system across the IPsec tunnel. • `Decrypted packets`—Total number of packets decrypted by the local system across the IPsec tunnel.

Table 153: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
`AH Statistics`	- `Input bytes`—Total number of bytes received by the local system across the IPsec tunnel. - `Output bytes`—Total number of bytes transmitted by the local system across the IPsec tunnel. - `Input packets`—Total number of packets received by the local system across the IPsec tunnel. - `Output packets`—Total number of packets transmitted by the local system across the IPsec tunnel.
`Errors`	- `AH authentication failures`—Total number of authentication header (AH) failures. An AH failure occurs when there is a mismatch of the authentication header in a packet transmitted across an IPsec tunnel. - `Replay errors`—Total number of replay errors. A replay error is generated when a duplicate packet is received within the replay window. - `ESP authentication failures`—Total number of Encapsulation Security Payload (ESP) failures. An ESP failure occurs when there is an authentication mismatch in ESP packets. - `ESP decryption failures`—total number of ESP decryption errors. - `Bad headers`—Total number of invalid headers detected. - `Bad trailers`—Total number of invalid trailers detected. - `Invalid SPI`— Total number of invalid SPIs packets detected. - `TS check fail`— Total number of TS check fail detected. - `Discarded`— Total number of discarded packets detected.

Sample Output

show security ipsec statistics

```
user@host> show security ipsec statistics
Virtual-system: Root
ESP Statistics:
  Encrypted bytes:                0
  Decrypted bytes:                0
  Encrypted packets:              0
  Decrypted packets:              0
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
  Invalid SPI: 0, TS check fail: 0
  Discarded: 0
```

Sample Output

show security ipsec statistics index 131073

```
user@host> show security ipsec statistics index 131073
ESP Statistics:
  Encrypted bytes:              952
  Decrypted bytes:              588
  Encrypted packets:              7
  Decrypted packets:              7
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
```

```
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
  Invalid SPI: 0, TS check fail: 0
  Discarded: 0

  FC Name     Encrypted Pkts  Decrypted Pkts  Encrypted bytes  Decrypted bytes
  best-effort 7                 7                 952              588
  custom_q1   0                 0                 0                  0
  custom_q2   0                 0                 0                  0
  network-control 0             0                 0                  0
  custom_q4   0                 0                 0                  0
  custom_q5   0                 0                 0                  0
  custom_q6   0                 0                 0                  0
  custom_q7   0                 0                 0                  0
  default     0                 0                 0                  0
```

Starting with Junos OS Release 18.2R1, the CLI `show security ipsec statistics index 131073` *index-number* output displays statistics for each forwarding class name.

Sample Output

show security ipsec statistics fpc 6 pic 1 (SRX Series Firewalls)

```
user@host> show security ipsec statistics fpc 6 pic 1
ESP Statistics:
Encrypted bytes:           536408
Decrypted bytes:           696696
Encrypted packets:           1246
Decrypted packets:            888
AH Statistics:
Input bytes:                    0
Output bytes:                   0
Input packets:                  0
Output packets:                 0
Errors:
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```

```
Invalid SPI: 0, TS check fail: 0
Discarded: 0
```

show security ipsec statistics ha-link-encryption (SRX5400, SRX5600, SRX5800)

Starting in Junos OS Release 20.4R1, when you configure the high availability (HA) feature, you can use this show command to view only interchassis link tunnel details. The following command displays only link encryption tunnel statistics on both nodes.

```
user@host> show security ipsec statistics ha-link-encryption
ESP Statistics:
  Encrypted bytes:            10376
  Decrypted bytes:             4996
  Encrypted packets:             96
  Decrypted packets:             96
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
  Invalid SPI: 0, TS check fail: 0
  Discarded: 0
```

show security ipsec statistics (MX-SPC3)

Starting with Junos OS Release 21.3R1, a new field **Tunnel MTU** in the output of the CLI `show security ipsec statistics` displays the option configured under `ipsec vpn hub-to-spoke-vpn tunnel-mtu` hierarchy.

```
user@host> show security ipsec statistics
  Encrypted bytes:                0
  Decrypted bytes:                0
  Encrypted packets:              0
  Decrypted packets:              0
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
  Invalid SPI: 0, TS check fail: 0
  Exceeds tunnel MTU: 0       --------  New counter
  Discarded: 0
```

```
user@host> show security ipsec statistics srg-id 1
ESP Statistics:
  Encrypted bytes:            10646
  Decrypted bytes:             4296
  Encrypted packets:             96
  Decrypted packets:             96
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
  Invalid SPI: 0, TS check fail: 0
  Exceeds tunnel MTU: 0
  Discarded: 0
```

Release Information

Command introduced in Junos OS Release 8.5. `fpc` and `pic` options added in Junos OS Release 9.3.

Support for the `ha-link-encryption` option added in Junos OS Release 20.4R1.

Support for the `srg-id` option added in Junos OS Release 22.4R1.

RELATED DOCUMENTATION

clear security ipsec statistics | 1701

show security ipsec traffic-selector

IN THIS SECTION

Syntax

```
show security ipsec traffic-selector interface-name interface-name
<brief | detail>
<destination-address address>
<fpc slot-number pic slot-number>
<kmd-instance (all | kmd-instance-name)>
<pic slot-number fpc slot-number>
<source-address address>
```

Description

Display information about the traffic selectors that have been negotiated between the initiator and responder.

Options

interface-name *interface-name* — Name of the secure tunnel logical interface.

`brief` | `detail` — (Optional) Display the specified level of output. The default is `brief`.

destination-address *address* — (Optional) Destination IP address.

fpc *slot-number* pic *slot-number* — (Optional) Display information about existing traffic selectors on the specified Flexible PIC Concentrator (FPC) slot and PIC slot.

kmd-instance — (Optional) Display information about existing traffic selectors in the key management process (in this case, it is KMD) identified by FPC slot-number and PIC slot-number. This option is used to filter the output.

- `all`—All KMD instances running on the Services Processing Unit (SPU).
- *`kmd-instance-name`*—Name of the KMD instance running on the SPU.

pic *slot-number* fpc *slot-number* — (Optional) Display information about existing traffic selectors on the specified PIC slot and FPC slot.

source-address *address* — (Optional) Source IP address.

Required Privilege Level

view

Output Fields

Table 154 on page 1948 lists the output fields for the `show security ipsec traffic-selector` command. Output fields are listed in the approximate order in which they appear.

Table 154: show security ipsec traffic-selector Output Fields

Field Name	Field Description	Level of Output
Tunnel-id	Tunnel ID.	All levels
Interface	Secure tunnel (st0) interface for the traffic selector.	All levels
IKE-ID	Peer IKE ID for the negotiated traffic selector.	All levels
Source IP	Source IP address for the negotiated traffic selector.	All levels
Destination IP	Destination IP address for the negotiated traffic selector.	All levels

Sample Output

show security ipsec traffic-selector interface-name st0.1 detail

```
user@host> show security ipsec traffic-selector interface-name st0.1 detail
Tunnel ID: 6920601, Interface: st0.1
IKE-ID: DC=Common_component, CN=enodeA, OU=Dept, O=Company, L=City, ST=CA, C=US
Source IP: ipv4 (192.0.2.0-192.0.2.255)
Destination IP: ipv4 (198.51.100.0-198.51.100.255)

Tunnel ID: 77594626, Interface: st0.1
IKE-ID: DC=Common_component, CN=enodeB, OU=Det, O=Company, L=City, ST=CA, C=US
Source IP: ipv4 (192.0.2.0-192.0.2.255)
Destination IP: ipv4 (203.0.113.0-203.0.113.255)
```

Release Information

Command introduced in Junos OS Release 12.3X48-D10.

RELATED DOCUMENTATION

IPsec Overview | 20

show security ipsec tunnel-distribution

IN THIS SECTION

Syntax

```
show security ipsec tunnel-distribution
<brief | summary | summary-cpuload>
<srg-id number>
```

Description

Display the number of IPsec VPN tunnels that are anchored in each thread. An IPsec tunnel session is assigned an anchor thread, based on the load during the tunnel session installation. When a new tunnel session is created, the least loaded thread is chosen to anchor the new tunnel. When the tunnel is deleted, the anchor mapping is removed from the control plane.

Tunnel distribution across different Services Processing Unit (SPU) or equivalent is based on the number of tunnels and not on throughput in each tunnel. Tunnels anchored in a SPU are not transferred to a different SPU or equivalent during SPU failure.

The distribution profile shows any assigned IPSec distribution profile without any distribution profiles assigned to a vpn object. This tab shows `default_profiile`, else the associated profile is displayed.

Options

none
Display thread information about all active tunnels.

`brief`
(Optional) Display thread information about all active tunnels. (Default)

`fpc`
FPC slot number (0..5).

`pic`
PIC slot number (0..3).

`summary`
(Optional) Display the number of tunnels anchored to each thread.

`summary-cpuload`
(Optional) Displays the load on each FPC and PIC. You can use this option to check the load on each FPC and PIC before or after redistributing the tunnel. See "show security ipsec tunnel-distribution summary-cpuload" on page 1954.

`srg-id`
(Optional) Display information related to a specific services redundancy group (SRG) in a Multinode High Availability setup.

Required Privilege Level

view

Output Fields

Table 155 on page 1951 lists the output fields for the `show security ipsec tunnel-distribution` command. Output fields are listed in the approximate order in which they appear.

Table 155: show security ipsec tunnel-distribution Output Fields

Field Name	Field Description	Level of Output
Tunnel-ID	VPN tunnel identifier.	`brief`
Thread-ID	Thread identifier.	All levels
Number of Tunnels	The number of tunnels anchored to the thread.	`summary`
CPU:1m	CPU load average for last 1 minute for FPC or PIC.	`summary-cpuload`
CPU:1h	CPU load average for last 1 hour for FPC or PIC.	`summary-cpuload`
CPU:1d	CPU load average for last 1 day for FPC or PIC.	`summary-cpuload`

Sample Output

show security ipsec tunnel-distribution

```
user@host> show security ipsec tunnel-distribution
Tunnel-ID          FPC        PIC       Thread-ID
------------------------------------------------------------------
  500006              0        1        4
  500012              0        1        8
  500009              0        1        6
  500002              0        1        1
  500005              0        1        3
  500001              0        0        15
  500008              0        1        5
  500010              0        0        18
  500004              0        0        16
  500003              0        1        2
  500011              0        1        7
  500007              0        0        17
```

```
Tunnel-ID                       FPC         PIC     Thread-ID  Distribution-profile
------------------------------------------------------------------
  500755                  0         1         1       spc-3
  500756                  2         0         0       spc-2
  500758                  0         1         1       default_profile
```

show security ipsec tunnel-distribution summary

```
user@host> show security ipsec tunnel-distribution summary
Number of Tunnels       FPC       PIC       Thread-ID
------------------------------------------------------------------
  1                       0         0         15
  1                       0         0         16
  1                       0         0         17
  1                       0         0         18
  1                       0         1         1
  1                       0         1         2
  1                       0         1         3
  1                       0         1         4
  1                       0         1         5
  1                       0         1         6
  1                       0         1         7
  1                       0         1         8
```

show security ipsec tunnel-distribution fpc 0 pic 0

```
user@host> show security ipsec tunnel-distribution fpc 0 pic 0
Tunnel-ID          FPC        PIC       Thread-ID
------------------------------------------------------------------
  500001              0        0        15
  500010              0        0        18
  500004              0        0        16
  500007              0        0        17
```

show security ipsec tunnel-distribution fpc 0 pic 1

```
user@host> show security ipsec tunnel-distribution fpc 0 pic 1
Tunnel-ID          FPC        PIC       Thread-ID
------------------------------------------------------------------
  500006             0        1         4
  500012             0        1         8
  500009             0        1         6
  500002             0        1         1
  500005             0        1         3
  500008             0        1         5
  500003             0        1         2
  500011             0        1         7
```

show security ipsec tunnel-distribution summary fpc 0 pic 0

```
user@host> show security ipsec tunnel-distribution summary fpc 0 pic 0
Number of Tunnels       FPC       PIC       Thread-ID
------------------------------------------------------------------
  1                       0         0         15
  1                       0         0         16
  1                       0         0         17
  1                       0         0         18
  0                       0         0         19
  0                       0         0         20
  0                       0         0         21
  0                       0         0         22
  0                       0         0         23
  0                       0         0         24
  0                       0         0         25
  0                       0         0         26
  0                       0         0         27
```

show security ipsec tunnel-distribution summary fpc 0 pic 1

```
user@host> show security ipsec tunnel-distribution summary fpc 0 pic 1
Number of Tunnels       FPC       PIC       Thread-ID
------------------------------------------------------------------
  1                       0         1         1
  1                       0         1         2
  1                       0         1         3
  1                       0         1         4
  1                       0         1         5
  1                       0         1         6
  1                       0         1         7
  1                       0         1         8
  0                       0         1         9
  0                       0         1         10
  0                       0         1         11
  0                       0         1         12
  0                       0         1         13
  0                       0         1         15
  0                       0         1         16
  0                       0         1         17
  0                       0         1         18
  0                       0         1         19
  0                       0         1         20
  0                       0         1         21
  0                       0         1         22
  0                       0         1         23
  0                       0         1         24
  0                       0         1         25
  0                       0         1         26
  0                       0         1         27
```

show security ipsec tunnel-distribution summary-cpuload

This command displays the same output as `show security ipsec tunnel-distribution summary`, but includes load averages (last 1 minute, 1 hour, and 1 day) of all threads for each FPC and PIC.

```
user@host> show security ipsec tunnel-distribution summary-cpuload

node0:

-------------------------------------------------------------------------------------------------
----------
   Number of Tunnels  FPC    PIC  Thread-ID  CPU:1m   CPU:1h  CPU:1d

-------------------------------------------------------------------------------------------------
----------
```

```
    1                 0       0     15          0        0       0
    1                 0       0     16          0        0       0
```

show security ipsec tunnel-distribution srg-id

```
user@host> show security ipsec tunnel-distribution srg-id 1
Tunnel-ID          FPC        PIC       Thread-ID Distribution Profile
------------------------------------------------------------------
  17277221           0         0        1        default-profile
```

Release Information

Command introduced in Junos OS Release 17.4R1.

summary-cpuload option introduced in Junos OS Release 20.4R1.

srg-id option introduced in Junos OS Release 22.4R1.

RELATED DOCUMENTATION

show security ipsec security-associations | 1899

request security re-distribution ipsec-vpn | 1767

show security re-distribution ipsec-vpn | 1980

show security ipsec tunnel-events-statistics

IN THIS SECTION

Syntax

```
show security ipsec tunnel-events-statistics
```

Description

Show tunnel event statistics.

Required Privilege Level

view

Sample Output

show security ipsec tunnel-events statistics

```
user@host> show security ipsec tunnel-events statistics
IPSec SA delete payload received from peer                                 : 153
Configuration change triggered clearing of IPSec SA                        : 1
Peer's remote IKE-ID validation failed during negotiation                  : 2
Phase1 proposal mismatch detected                                          : 2
Phase2 proposal mismatch detected                                          : 2
Peer proposed traffic-selectors are not in configured range                : 8576
Negotiation failed as peer did not respond                                 : 4
IKE SA negotiation successfully completed                                  : 19
IPSec SA negotiation successfully completed                                : 154
```

```
PKI validation failed: Peer's CA not configured in trusted-CA-group in IKE policy : 1
Tunnel is ready. Waiting for trigger event or peer to trigger negotiation  : 1
```

Release Information

Command introduced in Junos OS Release 12.3X48-D10.

Starting with Junos OS Release 15.1X49-D120, you can configure the CLI option `reject-duplicate-connection` at the [`edit security ike gateway` *`gateway-name`* `dynamic`] hierarchy level to retain an existing tunnel session and reject negotiation requests for a new tunnel with the same IKE ID. By default, an existing tunnel is tear down when a new tunnel with the same IKE ID is established. The `reject-duplicate-connection` option is only supported when `ike-user-type group-ike-id` or `ike-user-type shared-ike-id` is configured for the IKE gateway; the `aaa access-profile` *`profile-name`* configuration is not supported with this option.

Use the CLI option `reject-duplicate-connection` only when you are certain that reestablishment of a new tunnel with the same IKE ID should be rejected.

RELATED DOCUMENTATION

clear security ipsec tunnel-events-statistics | 1706

show security ipsec vpn vpnname ike idle time

IN THIS SECTION

Syntax

```
show security ipsec vpn vpn-nameike idle-time
```

Description

Displays the idle time to delete SAs.

Options

seconds The range of values, in seconds, is 0 through 999,999.

Additional Information

Required Privilege Level

view

Sample Output

show security ipsec vpn *vpn-name* ike idle-time

```
user@host> show security ipsec vpn vpn-name ike idle-time
```

Release Information

Command introduced in Junos OS Release 22.3R1.

show security pki ca-certificate (View)

IN THIS SECTION

Syntax

```
show security pki ca-certificate
<brief | detail>
<ca-profile ca-profile-name>
```

Description

Display information about the certificate authority (CA) public key infrastructure (PKI) digital certificates configured on the device.

The FIPS image does not permit the use of MD5 fingerprints. Therefore, MD5 fingerprints are not included when a certificate is displayed using this command. The SHA-1 fingerprint that is currently displayed is retained in the FIPS image. The Simple Certificate Enrollment Protocol (SCEP) is disabled in the FIPS image.

Options

- none—Display basic information about all configured CA certificates.
- `brief` | `detail`—(Optional) Display the specified level of output.
- `ca-profile` *`ca-profile-name`*- (Optional) Display information about only the specified CA certificate.

Required Privilege Level

view

Output Fields

Table 156 on page 1960 lists the output fields for the `show security pki ca-certificate` command. Output fields are listed in the approximate order in which they appear.

Table 156: show security pki ca-certificate Output Fields

Field Name	Field Description
`CA profile`	Name of the CA profile in the CA certificate. Starting in Junos OS Release 21.4R1, you can view this information by executing the `show security pki ca-certificate <brief | detail>` command.
`Certificate identifier`	Name of the digital certificate.
`Certificate version`	Revision number of the digital certificate.
`Serial number`	Unique serial number of the digital certificate.

Table 156: show security pki ca-certificate Output Fields *(Continued)*

Field Name	Field Description
`Issuer`	Authority that issued the digital certificate, including details of the authority organized using the distinguished name format. Possible subfields are: • `Organization`—Organization of origin. • `Organizational unit`—Department within an organization. • `Country`—Country of origin. • `Locality`—Locality of origin. • `Common name`—Name of the authority.
`Subject`	Details of the digital certificate holder organized using the distinguished name format. Possible subfields are: • `Organization`—Organization of origin. • `Organizational unit`—Department within an organization. • `Country`—Country of origin. • `Locality`—Locality of origin. • `Common name`—Name of the authority. If the certificate contains multiple subfield entries, all entries are displayed.
`Subject string`	Subject field as it appears in the certificate.
`Validity`	Time period when the digital certificate is valid. Values are: • `Not before`—Start time when the digital certificate becomes valid. • `Not after`—End time when the digital certificate becomes invalid.
`Public key algorithm`	Encryption algorithm used with the private key, such as `rsaEncryption(1024 bits)`.

Table 156: show security pki ca-certificate Output Fields *(Continued)*

Field Name	Field Description
`Signature algorithm`	Encryption algorithm that the CA used to sign the digital certificate, such as `sha1WithRSAEncryption`.
`Certificate Policy`	`Policy Identifier`—One or more policy object identifiers (OIDs).
`Use for key`	Use of the public key, such as `Certificate signing`, `CRL signing`, `Digital signature`, or `Data encipherment`.
`Fingerprint`	Secure Hash Algorithm (`SHA1`) and Message Digest 5 (`MD5`) hashes used to identify the digital certificate. Starting in Junos OS Release 21.4R1, you can also view the `SHA256` fingerprint for a local certificate along with `SHA1` and `MD5` fingerprints.
`Distribution CRL`	Distinguished name information and the URL for the certificate revocation list (CRL) server.

Sample Output

show security pki ca-certificate (MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

Starting in Junos OS Release 21.4R1, execute the `show security pki ca-certificate <ca-profile` *`ca-profile-name`*`>` command to view the CA profile name printed in the CA. The `CA profile` field in the output represents the CA profile name printed in the CA. In this sample, the CA profile name printed in the CA certificate is a `Root-CA`.

```
user@host> show security pki ca-certificate ca-profile Root-CA
LSYS: root-logical-system
  CA profile: Root-CA
Certificate identifier: Root-CA
  Issued to: Root-CA, Issued by: C = us, O = juniper, CN = Root-CA
  Validity:
    Not before: 05-19-2021 08:05 UTC
    Not after: 05-17-2031 08:05 UTC
  Public key algorithm: rsaEncryption(2048 bits)
  Keypair Location: Keypair generated locally
```

show security pki ca-certificate ca-profile detail (MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

Starting in Junos OS Release 21.4R1, execute the `show security pki ca-certificate <ca-profile` *ca-profile-name*`>` `detail` command to view:

- the CA profile name printed in the CA. The `CA profile` field in the output represents the CA profile name printed in the CA. In this sample, the CA profile name printed in the CA certificate is `Root-CA`.
- the SHA256 fingerprint for a CA certificate.

```
user@host> show security pki ca-certificate ca-profile Root-CA detail
LSYS: root-logical-system
  CA profile: Root-CA
Certificate identifier: Root-CA
  Certificate version: 3

  Serial number:
    hexadecimal: 0x00000d87
    decimal: 3463
  Issuer:
    Organization: juniper, Country: us, Common name: Root-CA
  Subject:
    Organization: juniper, Country: us, Common name: Root-CA
  Subject string:
    C=us, O=juniper, CN=Root-CA

  Validity:
    Not before: 05-19-2021 08:05 UTC
    Not after: 05-17-2031 08:05 UTC
  Public key algorithm: rsaEncryption(2048 bits)
    30:82:01:0a:02:82:01:01:00:cf:28:0c:04:ae:f0:89:f1:0a:cc:b3
    5a:0a:d9:c7:0a:f3:90:2e:7d:06:73:a4:65:94:3d:53:d4:25:2e:40
    11:98:4e:2f:52:53:1e:b3:69:2b:80:89:2e:b0:17:3a:3d:96:b3:70
    26:f7:da:ae:4e:ba:15:50:db:42:bd:bc:8c:0c:fd:5b:8e:f5:fb:74
    3c:48:8f:ec:c0:6a:5f:46:b3:1f:19:10:10:c4:e2:7e:e7:c5:ed:e1
    ff:64:01:01:f5:69:82:47:7a:2f:4c:6f:52:df:a4:06:fb:f8:ac:04
    3c:46:51:08:b4:5d:71:f3:69:a1:22:cb:53:18:74:bc:bf:4d:6b:4a
    b0:cd:4c:60:38:5f:ec:a8:6d:6c:77:dd:ed:14:a1:5f:c7:84:a7:74
    7a:6c:45:fa:4e:8a:db:8d:6c:ec:6a:25:fa:38:54:97:ac:0e:d0:12
    48:e5:0f:10:b2:3d:b0:de:95:53:d3:c8:a5:dc:6f:ed:f5:7d:49:e3
    b5:68:98:24:a7:8b:5d:a7:e5:98:de:51:b5:20:68:15:22:64:f1:c3
    cc:c4:1a:1a:be:bf:cb:fb:a7:79:92:a8:45:a3:ef:0d:2e:0f:21:f4
    5e:9d:77:1f:32:68:45:e1:93:ab:27:88:a6:c6:b2:81:55:a1:6d:c6
    81:85:1b:7f:61:02:03:01:00:01
  Signature algorithm: sha256WithRSAEncryption
  Distribution CRL:
    http://10.48.148.132:8080/crl-as-der/currentcrl-11.crl?id=11
  Authority Information Access OCSP:
    http://10.48.148.132:8090/Root-CA/
  Use for key: CRL signing, Certificate signing, Key encipherment, Digital signature
  Fingerprint:
    b4:65:6b:a2:28:01:b1:76:26:8b:8f:4f:53:b9:50:a6:eb:df:39:3a (sha1)
    14:c9:4f:da:96:15:94:6f:fa:5e:fd:60:ce:47:90:97 (md5)

49:ee:63:56:72:0b:f4:87:08:75:c9:1a:fa:6c:4d:c7:7c:2f:a2:21:31:68:30:67:87:37:cd:c0:86:34:1c:76
(sha256)
```

Release Information

Command modified in Junos OS Release 8.5.

Subject string output field added in Junos OS Release 12.1X44-D10. Policy identifier output field added in Junos OS Release 12.3X48-D10.

`CA profile` and `(sha256)` for `Fingerprint` output field added in Junos OS Release 21.4R1.

RELATED DOCUMENTATION

ca-profile (Security PKI)

request security pki ca-certificate verify (Security)

show security pki certificate-request (View)

IN THIS SECTION

Syntax

```
show security pki certificate-request
<brief|detail>
<certificate-id certificate-id-name>
```

Description

Display information about manually generated local digital certificate requests that are stored on the device.

Options

- none—Display basic information about all local digital certificate requests.
- `brief` / `detail`—(Optional) Display the specified level of output.

- `certificate-id` *`certificate-id-name`* —(Optional) Display information about only the specified local digital certificate requests.

Required Privilege Level

view

Output Fields

Table 157 on page 1966 lists the output fields for the `show security pki certificate-request` command. Output fields are listed in the approximate order in which they appear.

Table 157: show security pki certificate-request Output Fields

Field Name	Field Description
`Certificate identifier`	Name of the digital certificate.
`Certificate version`	Revision number of the digital certificate.
`Issued to`	Device that was issued the digital certificate.
`Subject`	Details of the digital certificate holder organized using the distinguished name format. Possible subfields are: • `Organization`—Organization of origin. • `Organizational unit`—Department within an organization. • `Country`—Country of origin. • `Locality`—Locality of origin. • `Common name`—Name of the authority.
`Alternate subject`	Domain name or IP address of the device related to the digital certificate.

Table 157: show security pki certificate-request Output Fields *(Continued)*

Field Name	Field Description
`Public key algorithm`	Encryption algorithm used with the private key, such as `rsaEncryption(1024 bits)`.
`Public key verification status`	Public key verification status: `Failed` or `Passed`. The `detail` output also provides the verification hash.
`Fingerprint`	Secure Hash Algorithm (`SHA1`) and Message Digest 5 (`MD5`) hashes used to identify the digital certificate.
`Use for key`	Use of the public key, such as `Certificate signing`, `CRL signing`, `Digital signature`, or `Data encipherment`.

Sample Output

show security pki certificate-request certificate-id user brief

```
user@host> show security pki certificate-request certificate-id hassan brief
Certificate identifier: user
      Issued to: user@example.net
      Public key algorithm: rsaEncryption(1024 bits)
```

Sample Output

show security pki certificate-request certificate-id user detail

```
user@host> show security pki certificate-request certificate-id hassan detail
    Certificate identifier: user
      Certificate version: 3
      Subject:
        Organization: example, Organizational unit: example, Country: IN,
                        Common name: user1
      Alternate subject: 192.168.72.124
      Public key algorithm: rsaEncryption(1024 bits)
      Public key verification status: Passed
        c7:a4:fb:e7:8c:4f:31:e7:eb:01:d8:32:65:21:f2:eb:6f:7d:49:1a:c3:9b
        63:47:e2:4f:f6:db:f6:c8:75:dd:e6:ec:0b:35:0a:62:32:45:6b:35:1f:65
        c9:66:b7:40:b2:f9:2a:ab:5b:60:f7:c7:73:36:da:68:25:fc:40:4b:12:3c
        d5:c8:c6:66:f6:10:1e:86:67:a8:95:9b:7f:1c:ae:a7:55:b0:28:95:a7:9a
        a2:24:28:e4:5a:b2:a9:06:7a:69:37:20:15:e1:b6:66:eb:22:b5:b6:77:f6
        65:88:b0:94:2b:91:4b:99:78:4a:e3:56:cc:14:45:d7:97:fd
      Fingerprint:
        8f:22:1a:f2:9f:27:b0:21:6c:da:46:64:31:34:1f:68:42:5a:39:e0 (sha1)
        09:15:11:aa:ea:f9:5a:b5:70:d7:0b:8e:be:a6:d3:cb (md5)
      Use for key: Digital signature
```

Release Information

Command modified in Junos OS Release 8.5.

RELATED DOCUMENTATION

clear security pki key-pair (Local Certificate) | 1707

show security pki crl (View)

IN THIS SECTION

Syntax

```
show security pki crl
                  <                brief                     /                 detail>
<ca-profile          ca-profile-name       >
```

Description

Display information about the certificate revocation lists (CRLs) configured on the device.

Options

- none—Display basic information about all CRLs.
- `brief` | `detail`—(Optional) Display the specified level of output.
- `ca-profile` *`ca-profile-name`*- (Optional) Display information about only the specified CA profile.

Required Privilege Level

view

Output Fields

Table 158 on page 1970 lists the output fields for the `show security pki crl` command. Output fields are listed in the approximate order in which they appear.

Table 158: show security pki crl Output Fields

Field Name	Field Description
`CA profile`	Name of the configured CA profile.
`CRL version`	Revision number of the certificate revocation list.
`CRL issuer`	Authority that issued the digital certificate, including details of the authority organized using the distinguished name format. Possible subfields are: • `emailAddress`—Mail address of the issuing authority. • `C`—Country of origin. • `ST`—State of origin. • `L`—Locality of origin. • `O`—Organization of origin. • `OU`—Department within an organization. • `CN`—Name of the authority.
`Effective date`	Date and time the certificate revocation list becomes valid.
`Next update`	Date and time the routing platform will download the latest version of the certificate revocation list.
`Revocation List`	List of digital certificates that have been revoked before their expiration date. Values are: • `Serial number`—Unique serial number of the digital certificate. • `Revocation date`—Date and time that the digital certificate was revoked.

Sample Output

show security pki crl ca-profile ca2

```
user@host> show security pki crl ca-profile ca2
CA profile: ca2
  CRL version: V00000001
  CRL issuer: emailAddress = user@example.net, C = US, ST = ca, L = sunnyvale, O = , OU = SPG
QA, CN = 2000-spg-example-net
  Effective date: 04-26-2007 18:47
  Next update: 05- 4-2007 07:07
```

Sample Output

show security pki crl ca-profile ca2 brief

```
user@host> show security pki crl ca-profile ca2 brief
CA profile: ca2
  CRL version: V00000001
  CRL issuer: emailAddress = user@example.net, C = US, ST = ca, L = sunnyvale, O = example
networks, OU = SPG QA, CN = 2000-spg-example-net
  Effective date: 04-26-2007 18:47
  Next update: 05- 4-2007 07:07
```

Sample Output

show security pki crl ca-profile ca2 detail

```
user@host> show security pki crl ca-profile ca2 detail
CA profile: ca2
  CRL version: V00000001
  CRL issuer: emailAddress = user@example.net, C = US, ST = ca, L = sunnyvale, O = example, OU =
SPG QA, CN = 2000-spg-example-net
  Effective date: 04-26-2007 18:47
  Next update: 05- 4-2007 07:07
  Revocation List:
    Serial number              Revocation date
    174e6399000000000506       03-16-2007 23:09
    174ef3f3000000000507       03-16-2007 23:09
    17529cd6000000000508       03-16-2007 23:09
    1763ac26000000000509       03-16-2007 23:09
    21904e570000000050a        03-16-2007 23:09
    2191cf790000000050b        03-16-2007 23:09
    21f10eb600000000050c       03-16-2007 23:09
    2253ca2a00000000050f       03-16-2007 23:09
    2478939b000000000515       03-16-2007 23:09
    24f35004000000000516       03-16-2007 23:09
    277ddfa8000000000517       03-16-2007 23:09
    277e97bd000000000518       03-16-2007 23:09
    27846a76000000000519       03-16-2007 23:09
    2785176f00000000051a       03-16-2007 23:09
```

Release Information

Command modified in Junos OS Release 8.5.

RELATED DOCUMENTATION

crl (Security) | 1472

show security pki local-certificate (View)

IN THIS SECTION

Syntax

```
show security pki local-certificate
<brief /detail>
<certificate-id certificate-id-name>
<system-generated>
```

Description

Display information about the local digital certificates, corresponding public keys, and the automatically generated self-signed certificate configured on the device.

Options

- none—Display basic information about all configured local digital certificates, corresponding public keys, and the automatically generated self-signed certificate.
- `brief` | `detail`—(Optional) Display the specified level of output.
- certificate-id *certificate-id-name* —(Optional) Display information about only the specified local digital certificates and corresponding public keys.
- `system-generated`—Display information about the automatically generated self-signed certificate.

Required Privilege Level

view

Output Fields

Table 159 on page 1974 lists the output fields for the `show security pki local-certificate` command. Output fields are listed in the approximate order in which they appear.

Table 159: show security pki local-certificate Output Fields

Field Name	Field Description
`Certificate identifier`	Name of the digital certificate.
`Certificate version`	Revision number of the digital certificate.
`Serial number`	Unique serial number of the digital certificate. Starting in Junos OS Release 20.1R1, PKI local certificate serial number is displayed with **0x** as prefix to indicate that the PKI local certificate is in the hexadecimal format. Starting in Junos OS Release 21.4R1, you can view the serial number of the digital certificate in both hexadecimal and decimal formats.
`Issued to`	Device that was issued the digital certificate.
`Issued by`	Authority that issued the digital certificate.

Table 159: show security pki local-certificate Output Fields *(Continued)*

Field Name	Field Description
`Issuer`	Authority that issued the digital certificate, including details of the authority organized using the distinguished name format. Possible subfields are: • `Organization`—Organization of origin. • `Organizational unit`—Department within an organization. • `Country`—Country of origin. • `Locality`—Locality of origin. • `Common name`—Name of the authority.
`LSYS`	Name of the logical systems.
`Subject`	Details of the digital certificate holder organized using the distinguished name format. Possible subfields are: • `Organization`—Organization of origin. • `Organizational unit`—Department within an organization. • `Country`—Country of origin. • `Locality`—Locality of origin. • `Common name`—Name of the authority. • `Serial number`—Serial number of the device. If the certificate contains multiple subfield entries, all entries are displayed.
`Subject string`	Subject field as it appears in the certificate.
`Alternate subject`	Domain name or IP address of the device related to the digital certificate. For multiple FQDNs, displays only the last FQDN details. Starting Junos OS Release 22.4R2, with multiple FQDNs, this option shows all domain names, IPv4 or IPv6 addresses and email addresses related to the digital certificate configured on the device.

Table 159: show security pki local-certificate Output Fields *(Continued)*

Field Name	Field Description
Cert-Chain	Starting in Junos OS Release 21.4R1, you can view the certificate chain for a given local certificate.
Validity	Time period when the digital certificate is valid. Values are: • Not before—Start time when the digital certificate becomes valid. • Not after—End time when the digital certificate becomes invalid.
Public key algorithm	Encryption algorithm used with the private key, such as rsaEncryption(1024 bits).
Public key verification status	Public key verification status: Failed or Passed. The detail output also provides the verification hash.
Signature algorithm	Encryption algorithm that the CA used to sign the digital certificate, such as sha1WithRSAEncryption.
Fingerprint	Secure Hash Algorithm (SHA1) and Message Digest 5 (MD5) hashes used to identify the digital certificate. Starting in Junos OS Release 21.4R1, you can also view the SHA-256 fingerprint for a local certificate along with SHA-1 and MD-5 fingerprints.
Distribution CRL	Distinguished name information and URL for the certificate revocation list (CRL) server.
Use for key	Use of the public key, such as Certificate signing, CRL signing, Digital signature, or Data encipherment.

Sample Output

show security pki local-certificate certificate-id hello

```
user@host> show security pki local-certificate certificate-id hello
LSYS: root-logical-system
Certificate identifier: hello
  Issued to: tc5-5-1, Issued by: DC = Juniper, CN = root-551-AAA
  Validity:
    Not before: 10-14-2021 21:41 UTC
    Not after: 02-13-2026 14:27 UTC
  Public key algorithm: rsaEncryption(1024 bits)
  Keypair Location: Keypair generated locally
```

show security pki local-certificate system-generated

```
user@host> show security pki local-certificate system-generated

LSYS: root-logical-system
Certificate identifier: system-generated
Issued to: 4a505bb373d7, Issued by: CN = 4a505bb373d7, CN = system generated, CN = self-signed
Validity:
Not before: 07-12-2019 22:23 UTC
Not after: 07-10-2024 22:23 UTC
Public key algorithm: rsaEncryption(2048 bits)
Keypair Location: Keypair generated locally
```

show security pki local-certificate system-generated detail

```
user@host> show security pki local-certificate system-generated detail
LSYS: root-logical-system
Certificate identifier: system-generated
  Certificate version: 3

  Serial number:
    hexadecimal: 0x23171f4f104463e2847bc792c39eb614
    decimal: 46643037698975347221422984685160412692
  Issuer:
    Common name: 4a505bb373d7, Common name: system generated, Common name: self-signed
  Subject:
    Common name: 4a505bb373d7, Common name: system generated, Common name: self-signed
  Subject string:
    CN=4a505bb373d7, CN=system generated, CN=self-signed

  Validity:
    Not before: 07-12-2019 22:23 UTC
    Not after: 07-10-2024 22:23 UTC
  Public key algorithm: rsaEncryption(2048 bits)
    30:82:01:0a:02:82:01:01:00:d5:7e:5e:7a:15:90:e3:23:07:8e:e3
    4b:40:0e:95:33:31:8c:17:0b:d1:78:48:2e:b5:e8:cb:44:03:f1:fd
    00:57:af:e9:d9:2c:78:96:04:37:3c:4a:65:d9:f1:fb:72:14:7f:b2
    d3:42:d3:84:be:e8:c5:6c:e2:f5:91:8a:41:02:30:a7:8b:2f:10:5e
    ab:5e:4e:d7:d6:f1:e7:ad:e3:6c:16:8d:6b:3c:0e:11:e9:26:8a:38
    99:78:0a:57:67:cc:0a:ea:fa:35:2b:f3:51:4e:cc:30:ee:e9:a7:0a
    26:14:42:fc:1b:22:ec:2d:0c:3b:10:d5:fb:e3:e6:ae:c6:cc:e7:de
    0f:cf:4d:a7:87:11:e1:4e:7f:33:69:c0:16:4e:80:c8:57:b4:9a:f8
    90:15:d8:e6:3e:06:7a:1c:a3:34:91:92:a6:88:9f:14:f5:89:39:da
    0f:88:1c:b0:bd:7d:46:23:b2:42:e8:6f:d2:34:9e:f2:bd:00:34:23
    99:4e:bb:39:0e:e4:bb:b2:9b:53:02:36:30:10:b7:28:e3:c4:8c:0e
    4c:fd:cf:4f:58:81:72:91:b4:82:18:cf:ba:f6:76:59:f2:d5:36:e1
    3a:29:20:72:02:5b:26:45:6f:92:0c:8e:dc:6c:d4:1c:78:55:db:66
    3a:e9:9a:9c:81:02:03:01:00:01
  Signature algorithm: sha256WithRSAEncryption
  Fingerprint:
    0b:08:f8:bc:c6:a3:c1:41:75:2b:48:da:5d:a7:0f:d8:99:45:cd:8a (sha1)
    8a:1b:b9:79:19:c6:c3:88:05:a8:05:28:3c:f2:b0:e9 (md5)

a3:9b:c1:c4:55:a8:f8:79:6f:a9:27:fc:f8:5a:af:45:37:dd:42:5f:2f:2b:bb:85:e3:f0:d7:99:9d:93:65:b1
(sha256)
```

show security pki local-certificate detail (MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

Starting in Junos OS Release 21.4R1, execute the `show security pki local-certificate detail` command to view:

- the CA certificate chain for a local certificate. The output field `cert-chain` displays the CA certificate chain.

if there is no certificate chain available for a given local certificate, then the `cert-chain` field displays the *Issuer/Root CA name*. If certificate chain exists, then `cert-chain` displays the Root-CA, followed by intermediate CA's.

- the local certificate serial number in both hexadecimal and decimal format.
- the SHA-256 fingerprint for a local certificate.

```
user@host> show security pki local-certificate certificate-id localcert-Sub11 detail
LSYS: root-logical-system
Certificate identifier: localcert-Sub11
  Certificate version: 3

  Serial number:
    hexadecimal: 0x0000202f
    decimal: 8239
  Issuer:
    Organization: juniper, Country: us, Common name: Sub11-CA
  Subject:
    Organizational unit: net_name, Common name: localcert-Sub11, Domain component: Juniper
  Subject string:
    DC=Juniper, CN=localcert-Sub11, OU=net_name
  Alternate subject: "localcert-Sub11@juniper.net", localcert-Sub11.juniper.net, 3.3.3.1, ipv6 
empty
  Cert-Chain: Root-CA , Sub1-CA , Sub11-CA
  Validity:
    Not before: 05-19-2021 16:30 UTC
    Not after: 05-17-2031 08:05 UTC
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:ae:16:b6:d7:72:34:9e:ef:4b:9b:e2:c8:d1
    8b:2a:e4:04:16:7a:06:ac:d6:be:96:e3:2f:2b:ac:b9:28:42:1b:c4
    ef:10:1e:7d:76:a5:8f:c4:fa:b5:b6:c1:7d:53:15:b7:85:f0:aa:4c
    af:9d:35:1e:06:dc:38:ce:40:70:b3:63:b9:4c:55:eb:ba:61:85:40
    71:32:ec:5a:3a:83:1f:e3:bf:0f:8d:cd:f7:29:44:e2:c6:a3:10:62
    bb:aa:f1:ae:cc:6e:ef:8a:4e:cc:03:cf:e9:35:c5:8f:7a:21:a9:ee
    9b:c1:2d:a3:7b:94:6f:db:2a:d7:01:0a:1c:1b:c3:02:03:01:00:01
  Signature algorithm: sha256WithRSAEncryption
  Distribution CRL:
    http://10.48.148.132:8080/crl-as-der/currentcrl-23.crl?id=23
  Authority Information Access OCSP:
    http://10.48.148.132:8090/Sub11-CA/
  Fingerprint:
    4b:04:da:b1:03:a6:a2:fc:24:d4:e3:ec:61:7a:d0:10:97:10:25:9e (sha1)
    e4:6a:3d:90:a1:a2:ec:5b:3b:de:c6:3f:16:1d:02:d5 (md5)

40:d3:95:c6:3c:5e:0e:cd:32:ca:63:76:e9:83:8e:ca:ec:8a:c7:0e:84:bb:e5:a5:bc:e4:25:0c:54:0c:23:51
(sha256)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Release Information

Command modified in Junos OS Release 9.1.

Subject string output field added in Junos OS Release 12.1X44-D10.

`Cert-Chain`, `hexadecimal` and `decimal` for `Serial Number`, `(sha256)` for `Fingerprint` output fields are added in Junos OS Release 21.4R1.

RELATED DOCUMENTATION

clear security pki local-certificate (Device) | **1708**

request security pki local-certificate generate-self-signed (Security) | **1754**

show security ipsec statistics (MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall) | **1997**

clear security pki statistics | **1711**

show security re-distribution ipsec-vpn

IN THIS SECTION

Syntax

```
show security re-distribution ipsec-vpn <gateway-name gateway-name>
```

Description

After executing `request security re-distribution ipsec-vpn gateway-name` *gateway-name* command, it may take some time to establish a new tunnel. This command displays the commands for which the tunnels are in the pending or awaiting state to get established.

Options

none	Display information for all the gateways.
gateway-name	(Optional) Display information for the specified gateway.

Required Privilege Level

view

Output Fields

Table 160 on page 1982 lists the output fields for the `show security re-distribution ipsec-vpn` command. Output fields are listed in the approximate order in which they appear.

Table 160: show security re-distribution ipsec-vpn Output Fields

Field Name	Field Description	Level of Output
`Gateway-name`	Name of the IKE gateway.	All levels
`FPC`	FPC slot number.	All levels
`PIC`	PIC slot number.	All levels
`Thread-id`	Thread identifier.	All levels
`Remote-id`	Remote identifier.	All levels

Sample Output

show security re-distribution ipsec-vpn

```
user@host> show security re-distribution ipsec-vpn

Gateway-name       FPC       PIC        Thread-id       Remote-id
Gateway-name-1      3         1            3               n/a
Gateway-name        2         2            5               n/a
Gateway-name-2      1         3            7             ike-id-3
```

Sample Output

show security re-distribution ipsec-vpn gateway-name

```
user@host> show security re-distribution ipsec-vpn gateway-name gateway-name-1

Gateway-name        FPC        PIC      Thread-id         Remote-id
Gateway-name-1       3          1            3             ike-id-1
Gateway-name-1       3          0            4             ike-id-3
```

Release Information

Command introduced in Junos OS Release 20.4R1.

RELATED DOCUMENTATION

request security re-distribution ipsec-vpn | 1767

show security ipsec tunnel-distribution | 1949

show security pki statistics

IN THIS SECTION

Syntax

```
show security pki statistics
```

Description

Display standard PKI statistics.

Options

None

Required Privilege Level

view

Output Fields

Table 161 on page 1984 lists the output fields for the `show security ipsec statistics` command.

Table 161: show security ipsec statistics Output Fields

Field Name	Field Description
`iked_msgs_inv`	Invalid messages from `iked` process.
`iked_msgs_rxd`	Messages received from `iked` process.
`iked_msgs_txd`	Messages sent to `iked` process.
`cc_kp_req`	Certificate chain keypair requests.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
`cc_kp_success`	Certificate chain keypair success.
`cc_kp_fail`	Certificate chain keypair fails (counter of no of certificate key-pair get failure).
`cc_id_ip`	Peer ID type is IP.
`cc_id_dn`	Peer ID type is DN (Domain Name).
`cc_id_fqdn`	Peer ID type is FQDN(Fully Qualified Domain Name).
`cc_id_user_fqdn`	User ID type is FQDN.
`cc_verify_req`	Number of certificate chain verification requests.
`cc_verify_success`	Number of successful certificate verifications.
`cc_verify_fail`	Number of failed certificate verifications
`cc_inv_ids`	IKE IDs did not match EE sub-alt-name .
`cc_inv_cert_count`	Invalid number of CA's in the certificate request.
`ocsp_requests_duplicate`	OCSP duplicate requests.
`ocsp_requests_sent`	OCSP requests sent.
`ocsp_resp_success`	Successful OCSP response.
`ocsp_resp_timeout`	OCSP response timed out.
`ocsp_action_fail`	OCSP next action failed on connection failure.
`ocsp_get_req_fail`	Failed to get OCSP request for a certificate.
`ocsp_resp_malformed_req`	Malformed OCSP response.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
`ocsp_resp_internal_error`	OCSP response has an internal error.
`ocsp_this_update_failed`	OCSP response is not valid yet.
`ocsp_next_update_failed`	Invalid next update time in OCSP response.
`ocsp_resp_try_later`	Busy OCSP responder or server. Try again later.
`ocsp_resp_sign_required`	OCSP responder requires signed request.
`ocsp_sign_verify_failed`	OCSP responder signature verification failed.
`ocsp_http_parse_error`	HTTP parsing error for OCSP response.
`ocsp_missing_cert_id`	OCSP response does not have responses for given certificate.
`ocsp_resp_unauthorized`	The OCSP responder does not accept requests from unauthorized clients.
`ocsp_rev_status_success`	OCSP certificate revocation check success.
`ocsp_rev_status_revoked`	OCSP certificate is revoked.
`ocsp_rev_status_unknown`	OCSP certificate revocation status is unknown.
`ocsp_nonce_check_failed`	Nonce check failed for OCSP responder.
`crl_entries_created`	Number of CRL entry created.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
`crl_entries_deleted`	Number of CRL entry deleted.
`mem_alloc_fails`	Memory allocation failure.
`crl_requests_sent`	Number of CRL requests sent.
`crl_responses_rcd`	Number of CRL responses received.
`crl_download_stop`	Number of CRL downloads stopped.
`crl_timer_start`	Number of times CRL timer started.
`crl_timer_stop`	Number of times CRL timer stopped.
`crl_revoked_certs`	Number of times certificates revoked due to CRl check.
`crl_revoke_skip`	Number of times CRL revocation check is skipped.
`crl_larger_size`	Received large CRL file greater than maximum file size limit.
`crl_download_failed`	Number of CRL download failures.
`crl_mem_alloc_fails`	Number of CRL entry memory allocation failures.
`crl_timer_mem_alloc_fails`	Number of CRL timer memory allocation failures.
`cmpv2_resp_invalid`	Number of Invalid CMPv2 responses.
`cmpv2_resp_invalid_status`	Failed to get valid CMPv2 response.
`cmpv2_resp_http_failed`	HTTP parsing failed for CMPv2 response.
`cmpv2_resp_validation_failed`	Number of CMPv2 response validation failures.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
`cmpv2_resp_null`	Number of NULL CMPv2 response received.
`cmpv2_resp_ca_cert_validation_failed`	Number of CMPv2 CA certificate validation success.
`cmpv2_resp_kup_ca_cert_missing`	CA certificate not found to validate CMPv2 response.
`cmpv2_resp_kup_ee_cert_missing`	EE or local certificate not found to validate CMPv2 response.
`cmpv2_resp_null_poll_resp`	CMPv2 poll-response is null.
`cmpv2_resp_no_trusted_ca`	Trusted CA is not available to validate received CA in CMPv2 response.
`cmpv2_resp_success`	Received valid CMPv2 response.
`cmpv2_ctx_set_caPubs_failed`	Failed to set ca-certificates received flag in CMPv2 context.
`cmpv2_ctx_set_extraCerts_failed`	Failed to set `extraCerts` field in CMPv2 context.
`cmpv2_load_local_failed`	CMPv2 local certificate load has failed.
`cmpv2_load_ca_failed`	CMPv2 CA certificate load has failed.
`cmpv2_poll_reached_max_retries`	No response from CMPv2 server after maximum configured retries.
`cmpv2_send_req_failed`	Failed to send CMPv2 requests.
`cmpv2_resp_nonce_check_failed`	CMPv2 responder nonce check failed.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
cmpv2_resp_stack_missing_issuer	Failed to get Issuer certificate for CMPv2 local certificate.
cmpv2_enroll_keypair_missing	CMPv2 Keypair does not exist for certificate.
cmpv2_auto_reenroll_new_keypair_missing	New key missing during CMPv2 auto-reenrollment.
cmpv2_auto_reenroll_keypair_missing	Key pair missing during CMPv2 auto-reenrollment.
cmpv2_auto_reenroll_cert_missing	Local certificate is missing during CMPv2 auto-reenrollment.
cmpv2_auto_reenroll_ca_profile_missing	CA profile configuration missing during CMPv2 auto-reenrollment.
cmpv2_send_http_req_failed	Failed to send CMPv2 HTTP request.
cmpv2_context_init_failed	CMPv2 context initialization failed.
cmpv2_context_search_failed	CMpv2 context search failed.
cmpv2_context_search_invalid_input	CMpv2 context search failed: due to invalid inputs.
cmpv2_context_create_invalid_input	CMPv2 context creation failed due to invalid inputs.
cmpv2_context_create_context_exists	CMPv2 context creation failed as CMPv2 context already exists.
cmpv2_context_freed	CMPv2 context freed.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
`cmpv2_gen_http_req_i2d_failed:`	CMPv2 message into DER format failed.
`cmpv2_gen_http_req_invalid_pkt_len`	CMPv2 HTTP request length is invalid.
`cmpv2_gen_http_req_failed`	Failed to generate CMPv2 HTTP request
`cmpv2_gen_http_req_invalid_msg_len`	Failed to generate CMPv2 HTTP request: invalid message length.
`cmpv2_search_timer_invalid_input`	Failed to get CMPv2 timer entry: invalid input.
`cmpv2_search_timer_failed`	Failed to get CMPv2 timer entry.
`cmpv2_stop_timer_failed`	Failed to stop CMPv2 timer.
`cmpv2_start_timer_failed`	Failed to start CMPv2 timer.
`cmpv2_send_message_failed`	Failed to send CMPv2 request to server.
`cmpv2_connection_failed`	Failed to connect to CMPv2 server.
`mem_alloc_failed`	pkid_malloc - failed to allocate memory.
`mem_alloc_type_invalid`	pkid_malloc - invalid type parameter.
`mem_free_type_invalid`	pkid_free - invalid type parameter.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
`mem_free_alloc_external`	pkid_free - not allocated by pkid_malloc.
`ldap_state_pending_release`	Pending LDAP state.
`ldap_state_released`	LDAP state is released or freed.
`scep_state_pending_release`	LDAP state needs to be released.
`scep_state_released`	SCEP state structure released or freed.
`scep_state_pkey3_initialised`	SCEP state keypair initialized.
`scep_state_pkey3_added`	Added SCEP state keypair.
`scep_state_pkey3_deleted`	Deleted SCEP state keypair.
`scep_ca_query_send_fail`	Failed to send SCEP request to server.
`scep_x509_lu_ca_obj_case:`	Received SCEP CA certificate case.
`scep_x509_lu_pkey_rs_ds_obj_case`	Received SCEP keypair case.
`scep_err_p_subject_is_null`	Missing subject in SCEP cert request.
`scep_p_err_keypair_is_null`	Keypair missing for certificate during SCEP process.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
scep_free_cert_req	Freed SCEP certificate request.
scep_reenroll_free_cert_req_info	Freed SCEP certificate request information during SCEP re-enrollment.
crl_state_pending_release	SCEP CRL check pending.
crl_state_released	SCEP CRL state freed.
ca_cert_issuer_verification_fail	Failed to CA certificate for given CA.
ae_cn_for_ca_cert_fail	Failed to get CA name for given CA certificate.
ae_cn_for_local_cert_fail	Failed to get CA name for given local certificate.
ae_get_cert_dn_fail	Failed to get subject DN field for given certificate id.
ae_x509_issuer_fail	Failed to get issuer certificate for given local certificate.
tpm_ae_key_null	TPM key is missing.
tpm_ae_key_gen_fail	TPM key generation failed.
tpm_key_gen_failure_uncaught	TPM key generation failure not captured.
pkid_db_open	PKI configuration DB is opened.
pkid_db_close	PKI configuration DB is closed
pkid_db_close_fail	Failed to close PKI configured DB.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
`tpm_ae_success_failure`	TPM: failed to store keypair to file.
`tpm_pkid_opendir_fail`	Failed to open keypair directory in case of TPM.
`hsm_session_create_success`	HSM session creation success.
`hsm_session_create_failure`	HSM session creation failure.
`hsm_key_create_success`	HSM key creation success.
`hsm_key_create_failure`	HSM key creation failed.
`hsm_key_sign_success`	HSM signature sign success.
`hsm_key_sign_failure`	HSM signature sign failed.
`hsm_cert_sign_verify_success`	HSM signature verification success.
`hsm_cert_sign_verify_failure`	HSM signature verification failed.
`hsm_pki_to_ike_success`	HSM keypair sent to iked process.
`hsm_pki_to_ike_failure`	HSM keypair sent to IKED failed.
`hsm_key_sign_verify_failure`	HSM: private key signing failed at HSM.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
`hsm_function_initialize_failure:`	HSM initialization function failed.
`hsm_pub_key_retrieval_failure`	HSM failed to retrieve public key.
`hsm_cleanup_failure`	HSM failed to cleanup data structures.
`hsm_session_sign_re_create_success`	Re-create HSM signature for given session.
`hsm_session_sign_re_create_failure`	Re-create HSM signature for the given session failed.
`hsm_ss_key_sign_success`	HSM self-signed key signature success.
`hsm_ss_key_sign_failure`	HSM self-signed key signature failure.
`hsm_ae_local_cert_delete_failure`	HSM local certificate deletion failure.
`hsm_ae_local_cert_verif_failure`	HSM local certificate verification failure.
`hsm_ss_cert_load_failure`	HSM failed to load the self-signed certificate.
`hsm_dummy_key_delete_fail`	HSM failed to create dummy keypair.
`pkid_ha_file_replicate_fail`	HSM failed to copy file to other node.
`pkid_mnha_ae_cert_load_fail`	MNHA certificate load failed.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
pkid_mnha_ae_cert_verification_fail	MNHA certificate verification failed.
mnha_file_sync_fail	MNHA failed to synchronize file to other node.
kqueue_init_error	kqueue initialization failure.
kqueue_cacert_hash_alloc_fail	kqueue failed to generate memory for CA certificate hash.
kqueue_cacert_file_open_fail	kqueue: failed to open CA certificate file.
kqueue_cacert_start_fail	kqueue failed.
kqueue_cacert_kevent_fail	kqueue: failed to add kevent.
kqueue_cacert_handler_register_fail	kqueue: CA certificate handler function failed.
kqueue_cacrl_hash_alloc_fail	kqueue: failed to allocate memory for CRL hash.
kqueue_cacrl_file_open_fail	kqueue: failed to open CRL file.
kqueue_cacrl_start_fail	kqueue: failed to get CRL.
kqueue_cacrl_kevent_fail	kqueue: failed to add kevent for CRL.
kqueue_cacrl_handler_register_fail	kqueue: CRL handler function failed.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
`kqueue_untrusted_ca_hash_alloc_fail`	kqueue: failed to allocate memory for untrusted CA certificate hash.
`kqueue_untrusted_ca_file_open_fail`	kqueue: failed to open untrusted CA certificate file.
`kqueue_untrusted_ca_start_fail`	kqueue failed for untrusted CA certificate.
`kqueue_untrusted_ca_kevent_fail`	kqueue failed to add untrusted CA certificate event .
`kqueue_untrusted_ca_handler_register_fail`	kqueue: untrusted CA handler function failed.
`kqueue_eecert_hash_alloc_fail`	kqueue: failed to allocate memory for local certificate hash.
`kqueue_eecert_file_open_fail`	kqueue: failed to open local certificate file.
`kqueue_eecert_start_fail`	kqueue: failed to get local certificate.
`kqueue_eecert_kevent_fail`	kqueue failed to add local certificate event.
`kqueue_eecert_handler_register_fail`	kqueue: local certificate handler function failed.
`kqueue_key_hash_alloc_fail`	kqueue: failed to allocate memory for keypair hash.
`kqueue_key_file_open_fail`	kqueue: failed to open keypair file.
`kqueue_key_start_fail`	kqueue: failed to get keypair.

Table 161: show security ipsec statistics Output Fields *(Continued)*

Field Name	Field Description
kqueue_key_kevent_fail	kqueue failed to add keypair kevent.
kqueue_key_handler_register_fail	kqueue: keypair handler function failed.
pkid_certchain_cacert_fail	Cannot find the signing certificate in the certificate store.
pkid_certs_less_than_min	The chain has less than two certificates. A chain must contain a minimum of two certificates.
pkid_untrust_certs_less_than_min	The untrusted certificate chain has less than two certificates.
pkid_ocsp_cert_issuer_null	OCSP failed to get the certificate issuer name.

Sample Output

show security ipsec statistics (MX240, MX480, MX960, SRX Series Firewalls and vSRX Virtual Firewall)

```
user@host> show security ipsec statistics
Statistic Name                 Value
--------------                 -----
iked_msgs_inv                    0
iked_msgs_rxd                    1862
iked_msgs_txd                    1869
cc_kp_req                        1862
cc_kp_success                    0
cc_kp_fail                       1862
cc_id_ip                         0
cc_id_dn                         0
cc_id_fqdn                       0
```

```
cc_id_user_fqdn                  0
cc_verify_req                    0
cc_verify_success                0
cc_verify_fail                   0
cc_inv_ids                       0
cc_inv_cert_count                0
ocsp_requests_duplicate          0
ocsp_requests_sent               0
ocsp_resp_success                0
ocsp_resp_timeout                0
ocsp_action_fail                 0
ocsp_get_req_fail                0
ocsp_resp_malformed_req          0
ocsp_resp_internal_error         0
ocsp_this_update_failed          0
ocsp_next_update_failed          0
ocsp_resp_try_later              0
ocsp_resp_sign_required          0
ocsp_sign_verify_failed          0
ocsp_http_parse_error            0
ocsp_missing_cert_id             0
ocsp_resp_unauthorized           0
ocsp_rev_status_success          0
ocsp_rev_status_revoked          0
ocsp_rev_status_unknown          0
ocsp_nonce_check_failed          0
crl_entries_created              0
crl_entries_deleted              0
mem_alloc_fails                  0
crl_requests_sent                0
crl_responses_rcd                0
crl_download_stop                0
crl_timer_start                  0
crl_timer_stop                   0
crl_revoked_certs                1
crl_revoke_skip                  0
crl_larger_size                  0
crl_download_failed              0
crl_mem_alloc_fails              0
crl_timer_mem_alloc_fails        0
cmpv2_resp_invalid               0
cmpv2_resp_invalid_status        0
cmpv2_resp_http_failed           0
```

```
cmpv2_resp_validation_failed     0
cmpv2_resp_null                  0
cmpv2_resp_ca_cert_validation_failed 0
cmpv2_resp_kup_ca_cert_missing   0
cmpv2_resp_kup_ee_cert_missing   0
cmpv2_resp_null_poll_resp        0
cmpv2_resp_no_trusted_ca         0
cmpv2_resp_success               0
cmpv2_ctx_set_caPubs_failed      0
cmpv2_ctx_set_extraCerts_failed  0
cmpv2_load_local_failed          0
cmpv2_load_ca_failed             0
cmpv2_poll_reached_max_retries   0
cmpv2_send_req_failed            0
cmpv2_resp_nonce_check_failed    0
cmpv2_resp_stack_missing_issuer  0
cmpv2_enroll_keypair_missing     0
cmpv2_auto_reenroll_new_keypair_missing 0
cmpv2_auto_reenroll_keypair_missing 0
cmpv2_auto_reenroll_cert_missing 0
cmpv2_auto_reenroll_ca_profile_missing 0
cmpv2_send_http_req_failed       0
cmpv2_context_init_failed        0
cmpv2_context_search_failed      0
cmpv2_context_search_invalid_input 0
cmpv2_context_create_invalid_input 0
cmpv2_context_create_context_exists 0
cmpv2_context_freed              0
cmpv2_gen_http_req_i2d_failed    0
cmpv2_gen_http_req_invalid_pkt_len 0
cmpv2_gen_http_req_failed        0
cmpv2_gen_http_req_invalid_msg_len 0
cmpv2_search_timer_invalid_input 0
cmpv2_search_timer_failed        0
cmpv2_stop_timer_failed          0
cmpv2_start_timer_failed         0
cmpv2_send_message_failed        0
cmpv2_connection_failed          0
cmpv2_ee_cert_get_keypair_failed 0
mem_alloc_failed                 0
mem_alloc_type_invalid           0
mem_free_type_invalid            0
mem_free_alloc_external          0
```

```
ldap_state_pending_release       0
ldap_state_released              0
scep_state_pending_release       0
scep_state_released              0
scep_state_pkey3_initialised     0
scep_state_pkey3_added           0
scep_state_pkey3_deleted         0
scep_ca_query_send_fail          0
scep_x509_lu_ca_obj_case         0
scep_x509_lu_pkey_rs_ds_obj_case 0
scep_err_p_subject_is_null       0
scep_p_err_keypair_is_null       0
scep_free_cert_req               0
scep_reenroll_free_cert_req_info 0
crl_state_pending_release        0
crl_state_released               0
ca_cert_issuer_verification_fail 0
ae_cn_for_ca_cert_fail           0
ae_cn_for_local_cert_fail        0
ae_get_cert_dn_fail              0
ae_x509_issuer_fail              0
tpm_ae_key_null                  0
tpm_ae_key_gen_fail              0
tpm_key_gen_failure_uncaught     0
pkid_db_open                     7
pkid_db_close                    7
pkid_db_close_fail               0
tpm_ae_success_failure           0
tpm_pkid_opendir_fail            0
hsm_session_create_success       0
hsm_session_create_failure       0
hsm_key_create_success           0
hsm_key_create_failure           0
hsm_key_sign_success             0
hsm_key_sign_failure             0
hsm_cert_sign_verify_success     0
hsm_cert_sign_verify_failure     0
hsm_pki_to_ike_success           0
hsm_pki_to_ike_failure           0
hsm_key_sign_verify_failure      0
hsm_function_initialize_failure  0
hsm_pub_key_retrieval_failure    0
hsm_cleanup_failure              0
```

```
hsm_session_sign_re_create_success 0
hsm_session_sign_re_create_failure 0
hsm_ss_key_sign_success          0
hsm_ss_key_sign_failure          0
hsm_ae_local_cert_delete_failure 0
hsm_ae_local_cert_verif_failure  0
hsm_ss_cert_load_failure         0
hsm_dummy_key_delete_fail        0
pkid_ha_file_replicate_fail      0
pkid_mnha_ae_cert_load_fail      0
pkid_mnha_ae_cert_verification_fail 0
mnha_file_sync_fail              0
kqueue_init_error                0
kqueue_cacert_hash_alloc_fail    0
kqueue_cacert_file_open_fail     0
kqueue_cacert_start_fail         0
kqueue_cacert_kevent_fail        0
kqueue_cacert_handler_register_fail 0
kqueue_cacrl_hash_alloc_fail     0
kqueue_cacrl_file_open_fail      0
kqueue_cacrl_start_fail          0
kqueue_cacrl_kevent_fail         0
kqueue_cacrl_handler_register_fail 0
kqueue_untrusted_ca_hash_alloc_fail 0
kqueue_untrusted_ca_file_open_fail 0
kqueue_untrusted_ca_start_fail   0
kqueue_untrusted_ca_kevent_fail  0
kqueue_untrusted_ca_handler_register_fail 0
kqueue_eecert_hash_alloc_fail    0
kqueue_eecert_file_open_fail     0
kqueue_eecert_start_fail         0
kqueue_eecert_kevent_fail        0
kqueue_eecert_handler_register_fail 0
kqueue_key_hash_alloc_fail       0
kqueue_key_file_open_fail        0
kqueue_key_start_fail            0
kqueue_key_kevent_fail           0
kqueue_key_handler_register_fail 0
pkid_certchain_cacert_fail       0
pkid_certs_less_than_min         0
pkid_untrust_certs_less_than_min 0
pkid_ocsp_cert_issuer_null       0
```

Release Information

Command introduced in Junos OS Release 21.4R1.

RELATED DOCUMENTATION

clear security pki statistics | 1711

show security tcp-encap connection

IN THIS SECTION

Syntax

```
show security tcp-encap connection
<brief | detail>
<session-id session-id>
```

Description

Display information about TCP encapsulation sessions.

Options

none	Display information about TCP encapsulation sessions.
brief | detail	(Optional) Display the specified level of output.
session-id *session-id*	(Optional) Display information for the specified session identifier.

Required Privilege Level

view

Output Fields

Table 162 on page 2003 lists the output fields for the `show security tcp-encap connection` command. Output fields are listed in the approximate order in which they appear.

Table 162: show security tcp-encap connection Output Fields

Field Name	Field Description
Session-Id	Session identifier.
Client	Name of the remote access client.
Gateway	IP address of the remote gateway.
Local Gateway	IP address of the local gateway.
Remote Gateway	IP address of the remote gateway.
Started	Date and time the connection started.

Table 162: show security tcp-encap connection Output Fields *(Continued)*

Field Name	Field Description
Anchor spu	Services Processing Unit (SPU) on which the connection is anchored.

Sample Output

show security tcp-encap connection

```
user@host> show security tcp-encap connection
Session-Id    Client            Gateway
   34         NCP-1              10.4.0.1
   644        NCP-1              10.5.0.1
```

show security tcp-encap connection detail

```
user@host> show security tcp-encap connection detail
Session id: 34
    Local Gateway: 10.4.0.2:500 , Remote Gateway: 10.4.0.1:9500
    Client: NCP-1
    Started: Sun Jan 08 2017 21:32:58
    Anchor spu: 1

Session id: 644
    Local Gateway: 10.4.0.2:443 , Remote Gateway: 10.5.0.1:9500
    Client: NCP-1
    Started: Sun Jan 08 2017 21:32:58
    Anchor spu: 1
```

show security tcp-encap connection session-id 644

```
user@host> show security tcp-encap connection session-id 644
Session id: 644
    Local Gateway: 10.4.0.2:443 , Remote Gateway: 10.5.0.1:9500
    Client: NCP-1
    Started: Sun Jan 08 2017 21:32:58
    Anchor spu: 1
```

Release Information

Command introduced in Junos OS Release 15.1X49-D80.

RELATED DOCUMENTATION

tcp-encap | 1635

show security tcp-encap statistics

IN THIS SECTION

Syntax

```
show security tcp-encap statistics
```

Description

Display TCP encapsulation statistics.

Required Privilege Level

view

Output Fields

Table 163 on page 2006 lists the output fields for the `show security tcp-encap statistics` command. Output fields are listed in the approximate order in which they appear.

Table 163: show security tcp-encap statistics Output Fields

Field Name	Field Description
Policy Matched	Number of policies matched.
TCP sessions	Number of TCP sessions.

Sample Output

show security tcp-encap statistics

```
user@host> show security tcp-encap statistics
TCP encapsulation statistics:
  Policy Matched:                    16
  TCP sessions:                      16
```

Release Information

Command introduced in Junos OS Release 15.1X49-D80.

RELATED DOCUMENTATION

clear security tcp-encap statistics | **1716**