Junos® OS

Public Key Infrastructure Feature Guide for Security Devices

Release

12.1X46-D10

Modified: 2016-07-07

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.

Junos® OS Public Key Infrastructure Feature Guide for Security Devices
12.1X46-D10

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

Table of Contents

Part 3 Administration

Part 4 Index

List of Figures

Part 1 Overview

Part 2 Configuration

List of Tables

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- J Series
- SRX Series
- LN Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

   ```
   [edit]
   user@host# edit system scripts
   [edit system scripts]
   ```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

   ```
   [edit system scripts]
   user@host# load merge relative /var/tmp/ex-script-snippet.conf
   load complete
   ```

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

Table 1 on page xiii defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
i	Informational note	Indicates important features or instructions.
⚠	Caution	Indicates a situation that might result in loss of data or hardware damage.
⚡	Warning	Alerts you to the risk of personal injury or death.
✳	Laser warning	Alerts you to the risk of personal injury from a laser.
💡	Tip	Indicates helpful information.
🏅	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xiv defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		    route default {
		      nexthop *address*;
		      retain;
		    }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

GUI Conventions

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.
- E-mail—Send your comments to **techpubs-comments@juniper.net**. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

Certificates and PKI

Understanding Certificates and PKI

Supported Platforms J Series, LN Series, SRX Series

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 Public Key Infrastructure (PKI) provides an infrastructure for digital certificate management.

This topic includes the following sections:

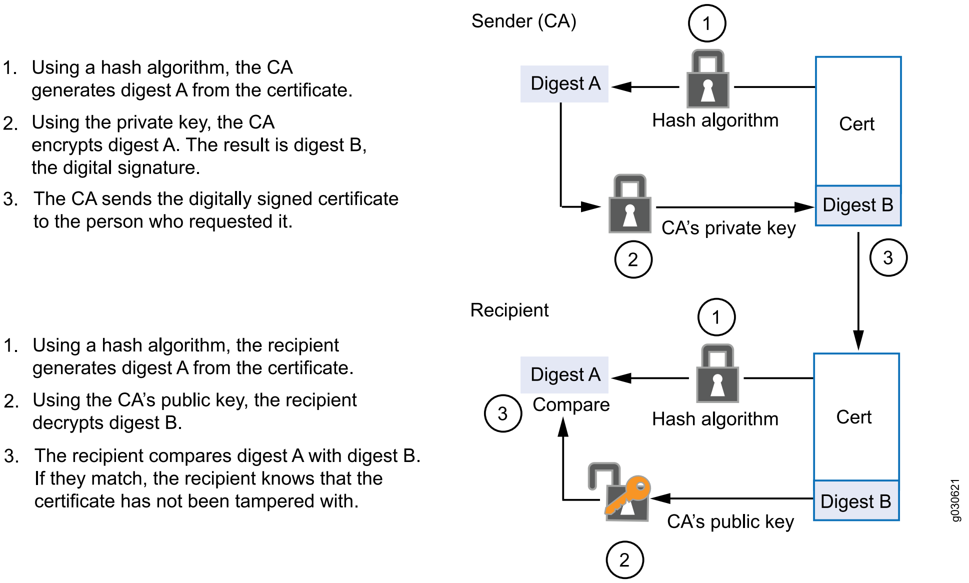
Certificate Signatures and Verification

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 4 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 4 illustrates this process.

NOTE: 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.

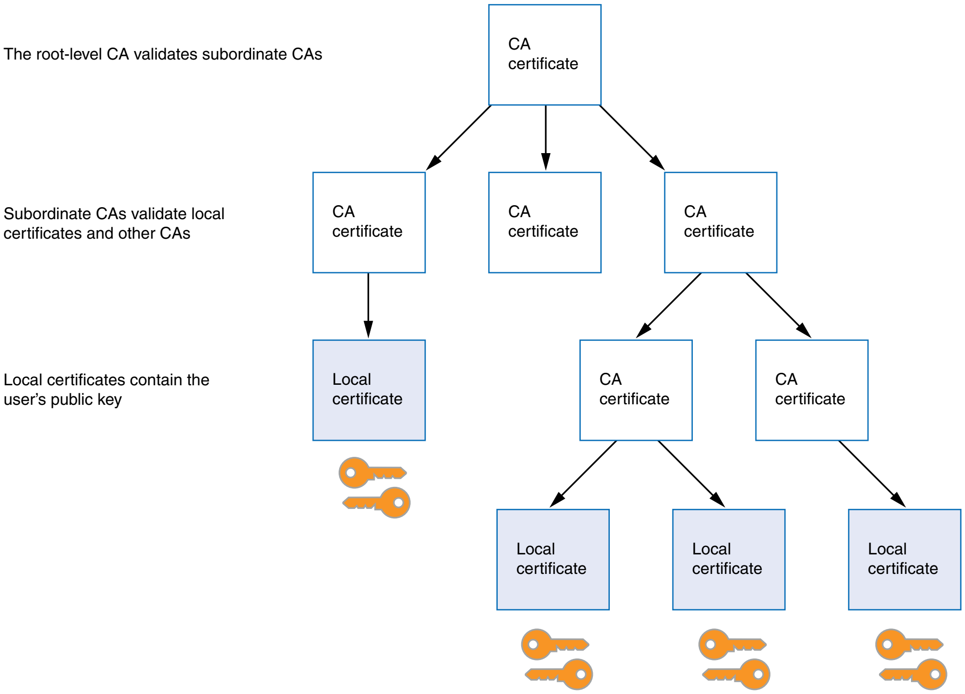
When Digital Signature Algorithm (DSA) signatures are used, the SHA-1 hash algorithm is used to generate the digest. When Rivest-Shamir-Adleman (RSA) signatures are used, SHA-1 is the default hash algorithm used to generate the digest; you can specify the SHA-256 hash algorithm with the **digest** option 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.

Public Key Infrastructure

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.

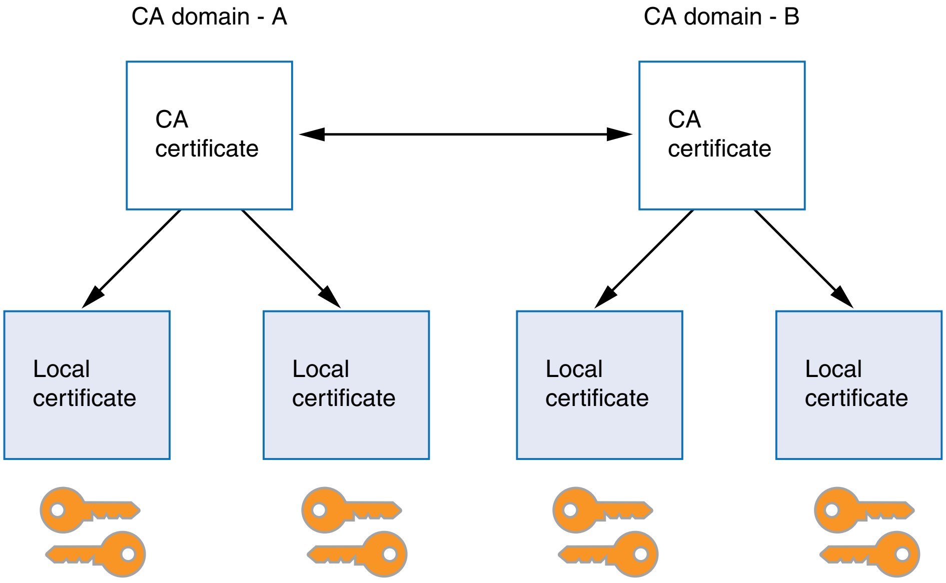
Figure 2 on page 5 shows the structure of a single-domain certificate authority with multiple hierarchy levels.

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 6.

Figure 3: Cross-Certification



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.

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:

- 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 supports automatic sending of certificate requests through the Simple Certificate Enrollment Protocol (SCEP).**

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

For convenience and practicality, PKI must be transparently managed and implemented. Toward this goal, Junos OS supports the following features:

- Generates a public-private key pair.
- Loads multiple local certificates from different CAs.
- Delivers a certificate when establishing an IPsec tunnel.
- Validates a certificate path upward through a single level of CA authorities.
- Supports the Public-Key Cryptography Standards #7 (PKCS #7) cryptographic . As a result, the device can accept X.509 certificates and certificate revocation lists (CRLs) packaged within a PKCS #7 envelope.

  NOTE: **Junos OS supports a PKCS #7 file size of up to 7 KB.**

- Retrieves CRLs online retrieval through Lightweight Directory Access Protocol (LDAP) or Hypertext Transfer Protocol (HTTP).

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.

Related Documentation

- Digital Certificates Configuration Overview on page 25
- Understanding Certificate Chains on page 19
- *VPN Overview*
- *Public Key Infrastructure Feature Guide for Security Devices*

Cryptographic Key Handling Overview

Supported Platforms J Series, LN Series, SRX Series

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 defence, may 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.

NOTE: 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 like configuring periodic self-test 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

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 2

Public-Private Key Pairs

Understanding Public Key Cryptography

Supported Platforms J Series, LN Series, SRX Series

The public-private key pairs used in public key cryptography play an important role in the use of digital certificates. A public-private key pair encrypts and decrypts data. Data encrypted with a public key, which the owner makes available to the public, can be decrypted with the corresponding private key only, which the owner keeps secret and protected. For example, if Alice wants to send Bob an encrypted message, Alice can encrypt it with Bob's public key and send it to him. Bob then decrypts the message with his private key.

The reverse process is also useful: encrypting data with a private key and decrypting it with the corresponding public key. This process is known as creating a digital signature. For example, if Alice wants to present her identity as the sender of a message, she can encrypt the message with her private key and send the message to Bob. Bob then decrypts the message with Alice's public key, thus verifying that Alice is indeed the sender.

When you generate a public-private key pair, the device automatically saves the key pair in a file in the certificate store, where it is subsequently used in certificate request commands. The generated key pair is saved as *certificate-id*.

NOTE: The default RSA and DSA key size is 1024 bits. If you are using the Simple Certificate Enrollment Protocol (SCEP), Junos OS supports RSA only.

NOTE: If the device renews a great number of certificates at once, thus using up keys rapidly, it might run out of pregenerated keys and have to generate them promptly for each new request. In this case, the generation of keys might affect the performance of the device, especially in a high-availability environment where the performance of the device might slow down for a number of minutes.

Related Documentation

- Understanding Certificates and PKI on page 3
- Example: Generating a Public-Private Key Pair on page 27
- Digital Certificates Configuration Overview on page 25
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 3

Certificate Authority Profiles

Understanding Certificate Authority Profiles

Supported Platforms J Series, LN Series, SRX Series

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

ℹ NOTE: The following CAs are supported: Entrust, Microsoft, and Verisign. SCEP only supports the Microsoft CA.

Related Documentation

- Understanding Certificates and PKI on page 3
- Example: Configuring a CA Profile on page 29
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 4

Certificates

Understanding Online CA Certificate Enrollment

Supported Platforms J Series, LN Series, SRX Series

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.

Related Documentation

- Understanding Public Key Cryptography on page 9
- Understanding Certificates and PKI on page 3
- Enrolling a CA Certificate Online Using SCEP on page 31
- Example: Enrolling a Local Certificate Online Using SCEP on page 32
- *Public Key Infrastructure Feature Guide for Security Devices*

Understanding Local Certificate Requests

Supported Platforms J Series, LN Series, SRX Series

When you create a local certificate request, the device generates a CA 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)

NOTE: 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.

Related Documentation

- Understanding Certificates and PKI on page 3
- Example: Manually Generating a CSR for the Local Certificate and Sending it to the CA Server on page 35
- *Public Key Infrastructure Feature Guide for Security Devices*

Understanding Certificate Loading

Supported Platforms J Series, LN Series, SRX Series

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.

NOTE: You can load multiple EE certificates onto the device.

Related Documentation

- Understanding Certificates and PKI on page 3
- Example: Loading CA and Local Certificates Manually on page 36
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 5

Certificate Revocation

Understanding Certificate Revocation Lists

Supported Platforms J Series, LN Series, SRX Series

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, participants check the CRL list to see if certificates received during an IKE exchange are still valid. If a CRL did not accompany a CA certificate and is not loaded on the device, the device tries to download it automatically from the CRL distribution point of the local certificate. If the device fails to connect to the URL in the certificate distribution point (CDP), it tries to retrieve the CRL from the URL configured in the CA profile.

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.

Related Documentation

- Understanding Certificates and PKI on page 3
- Example: Manually Loading a CRL onto the Device on page 67
- Example: Verifying Certificate Validity on page 70
- Deleting a Loaded CRL (CLI Procedure) on page 71
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 68
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 6

Self-Signed Certificates

Understanding Self-Signed Certificates

Supported Platforms J Series, LN Series, SRX Series

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.

NOTE: 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.

This topic includes the following sections:

Generating Self-Signed Certificates

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.

Automatically Generating Self-Signed Certificates

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.

Manually Generating Self-Signed Certificates

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.

Related Documentation

- Understanding Certificates and PKI on page 3
- Using Automatically Generated Self-Signed Certificates (CLI Procedure) on page 74
- Example: Manually Generating Self-Signed Certificates on page 73
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 7

Certificate Chains

Understanding Certificate Chains

Supported Platforms LN Series, SRX Series

Multilevel Hierarchy for Certificate Authentication

Certificate-based authentication is an authentication method supported on SRX Series devices 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.

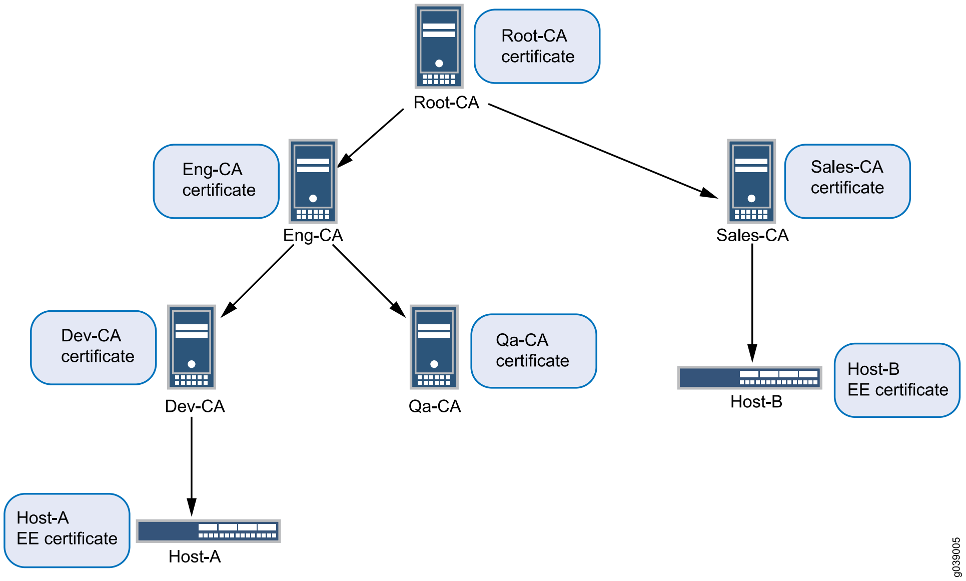
When a single-level hierarchy for certificate-based authentication is employed, all EE certificates in the network must be signed by the same CA. All firewall devices must have the same CA certificate enrolled for peer certificate validation. The certificate payload sent during IKE negotiation only contains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

In the example CA hierarchy shown in Figure 4 on page 20, 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 4: 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/”;
        }
    }
}
```

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.

A VPN device must be able to check a peer’s certificate for its revocation status. A device can use the CA certificate received from its peer to extract the URL to dynamically download the CA’s CRL and check the revocation status of the peer’s certificate. 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 CRL from the peer’s CA and check the revocation status of the peer’s certificate. In Figure 4 on page 20, 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.

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

The properties of a parent CA profile are inherited for dynamic CA profiles. In Figure 4 on page 20, 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 parent 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.

NOTE: 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. Because the CA certificate is not enrolled by an administrator, it is used only for validating the CRL received from the CA server and not for validating the peer certificate.

Related Documentation

- Example: Configuring a Device for Peer Certificate Chain Validation on page 77
- Understanding Certificates and PKI on page 3
- Understanding Certificate Authority Profiles on page 11
- Understanding Certificate Revocation Lists on page 15

PART 2

Configuration

CHAPTER 8

Certificates and PKI

Digital Certificates Configuration Overview

Supported Platforms J Series, LN Series, SRX Series

You can obtain CA and local certificates manually, or online using the Simple Certificate Enrollment Protocol (SCEP). Certificates are verifiable and renewable, and you can delete them when they are no longer needed.

Junos OS Release 8.5 and earlier support only manual certificate requests. This process includes generation of a PKCS10 request, submission to the CA, retrieval of the signed certificate, and manually loading of the certificate into the Juniper Networks device.

Automatic sending of certificate requests through SCEP is supported only in Junos OS Release 9.0 or later.

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.

This topic includes the following sections:

- Enabling Digital Certificates Online: Configuration Overview on page 25
- Manually Generating Digital Certificates: Configuration Overview on page 26

Enabling Digital Certificates Online: Configuration Overview

Supported Platforms J Series, LN Series, SRX Series

SCEP uses the online method to request digital certificates. To obtain a certificate online:

1. Generate a key pair on the device. See "Example: Generating a Public-Private Key Pair" on page 27.
2. Create a CA profile or profiles containing information specific to a CA. See "Example: Configuring a CA Profile" on page 29.
3. Enroll the CA certificate. See "Enrolling a CA Certificate Online Using SCEP" on page 31.
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 32.
5. Configure automatic reenrollment. See "Example: Using SCEP to Automatically Renew a Local Certificate" on page 34.

Manually Generating Digital Certificates: Configuration Overview

Supported Platforms J Series, LN Series, SRX Series

To obtain digital certificates manually:

1. Generate a key pair on the device. See "Example: Generating a Public-Private Key Pair" on page 27.
2. Create a CA profile or profiles containing information specific to a CA. See "Example: Configuring a CA Profile" on page 29.
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 35.
4. Load the certificate onto the device. See "Example: Loading CA and Local Certificates Manually" on page 36.
5. Configure automatic reenrollment. See "Example: Using SCEP to Automatically Renew a Local Certificate" on page 34.
6. If necessary, load the certificate's CRL on the device. See "Example: Manually Loading a CRL onto the Device" on page 67.
7. If necessary, configure the CA profile with CRL locations. See "Example: Configuring a Certificate Authority Profile with CRL Locations" on page 68

Related Documentation

- Understanding Certificates and PKI on page 3
- Example: Verifying Certificate Validity on page 70
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 68
- Deleting Certificates (CLI Procedure) on page 38
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 9

Public-Private Key Pairs

Example: Generating a Public-Private Key Pair

Supported Platforms J Series, LN Series, SRX Series

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

Step-by-Step Procedure

To generate a public-private key pair:

1. Create a certificate key pair.

```
[edit]
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
```

Related Documentation

- Understanding Public Key Cryptography on page 9

- Example: Verifying Certificate Validity on page 70
- Digital Certificates Configuration Overview on page 25
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 10

Certificate Authority Profiles

Example: Configuring a CA Profile

Supported Platforms J Series, LN Series, SRX Series

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. 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

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
     revocation-check crl refresh-interval 48 url http://www.my-ca.com/my-crl.crl
   ```

2. Specify the enrollment retry value.

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

3. 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
```

4. 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.

Related Documentation

- Understanding Certificate Authority Profiles on page 11
- Digital Certificates Configuration Overview on page 25
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 11

Certificates

Enrolling a CA Certificate Online Using SCEP

Supported Platforms J Series, LN Series, SRX Series

Before you begin:

1. Generate a public and private key pair. See "Example: Generating a Public-Private Key Pair" on page 27.
2. Create a CA profile. See "Example: Configuring a CA Profile" on page 29.

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**.

Related Documentation

- Understanding Online CA Certificate Enrollment on page 13
- Digital Certificates Configuration Overview on page 25
- Example: Enrolling a Local Certificate Online Using SCEP on page 32
- Example: Using SCEP to Automatically Renew a Local Certificate on page 34
- *Public Key Infrastructure Feature Guide for Security Devices*

Example: Enrolling a Local Certificate Online Using SCEP

Supported Platforms J Series, LN Series, SRX Series

This example shows how to enroll a local certificate online.

- Requirements on page 32
- Overview on page 32
- Configuration on page 33
- Verification on page 33

Requirements

Before you begin:

- Generate a public and private key pair. See "Example: Generating a Public-Private Key Pair" on page 27.
- Configure a certificate authority profile. See "Example: Configuring a CA Profile" on page 29.
- Enroll the CA certificate. See "Enrolling a CA Certificate Online Using SCEP" on page 31.

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 CA profile name as ca-profile-ipsec and the CA location as http://10.155.8.1/certsrv/mscep/mscep.dll.

You will use the **request security pki local-certificate enroll** command to start the online enrollment for the specified certificate ID. 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:

NOTE: SCEP sends a PKCS #10 format certificate request enveloped in PKCS #7 format.

- The challenge CA password for certificate enrollment and revocation—for example, **aaa**. If the CA does not provide the challenge password, then choose your own password.

- 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

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
  http://10.155.8.1/certsrv/mscep/mscep.dll
```

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 ca-profile ca-profile-ipsec
  certificate-id qqq challenge-password aaa 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
```

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

Verification

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

Related Documentation

- Understanding Online CA Certificate Enrollment on page 13
- Digital Certificates Configuration Overview on page 25
- Enrolling a CA Certificate Online Using SCEP on page 31
- Example: Using SCEP to Automatically Renew a Local Certificate on page 34

- *Public Key Infrastructure Feature Guide for Security Devices*

Example: Using SCEP to Automatically Renew a Local Certificate

Supported Platforms J Series, LN Series, SRX Series

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 “Enabling Digital Certificates Online: Configuration Overview” on page 25.
- Obtain a local certificate. See “Example: Enrolling a Local Certificate Online Using SCEP” on page 32.

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%, then configure 10 for the reenrollment trigger.

For this feature to work, the device must be able to reach the SCEP 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.

In this example, you can enable and disable automatic SCEP certificate renewal either for all SCEP certificates or on a per-certificate basis. You set the **security pki auto-re-enrollment** command to enable and configure certificate reenrollment. You specify the certificate ID of the CA certificate as sm1 and set the CA profile name associated with the certificate to aaa. You set the challenge password for CA certificate to abc. 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, by default, the Juniper Networks device uses the existing key pair. To generate a new key pair, use the **re-generate-keypair** command.

Configuration

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 certificate-id ca-ipsec
  ca-profile-name ca-profile-ipsec challenge-password abc
  re-enroll-trigger-time-percentage 10 re-generate-keypair
```

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.

Related Documentation

- Understanding Online CA Certificate Enrollment on page 13
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 68
- Enrolling a CA Certificate Online Using SCEP on page 31
- Example: Enrolling a Local Certificate Online Using SCEP on page 32
- *Public Key Infrastructure Feature Guide for Security Devices*

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

Supported Platforms J Series, LN Series, SRX Series

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

- Requirements on page 35
- Overview on page 36
- Configuration on page 36
- Verification on page 36

Requirements

Generate a public and private key. See “Example: Generating a Public-Private Key Pair” on page 27.

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

Step-by-Step Procedure

To generate a local certificate manually:

1. 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)
```

Related Documentation

- Understanding Local Certificate Requests on page 13
- Digital Certificates Configuration Overview on page 25
- *Public Key Infrastructure Feature Guide for Security Devices*

Example: Loading CA and Local Certificates Manually

Supported Platforms J Series, LN Series, SRX Series

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

- Requirements on page 37
- Overview on page 37
- Configuration on page 37
- Verification on page 37

Requirements

Before you begin:

- Generate a public-private key pair. See “Example: Generating a Public-Private Key Pair” on page 27.
- Create a CA profile. See “Understanding Certificate Authority Profiles” on page 11.

NOTE: 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 35.

Overview

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

Configuration

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 say 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
```

Related Documentation

- Understanding Certificate Loading on page 14
- Digital Certificates Configuration Overview on page 25
- Example: Using SCEP to Automatically Renew a Local Certificate on page 34
- Example: Verifying Certificate Validity on page 70
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 68
- *Public Key Infrastructure Feature Guide for Security Devices*

Deleting Certificates (CLI Procedure)

Supported Platforms J Series, LN Series, SRX Series

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.

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

Related Documentation

- Digital Certificates Configuration Overview on page 25
- *Public Key Infrastructure Feature Guide for Security Devices*

Example: Configuring PKI

Supported Platforms LN Series

This example shows how to configure, verify, and troubleshoot the PKI. This topic includes the following sections:

Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.4 or later
- SRX Series devices or J Series devices

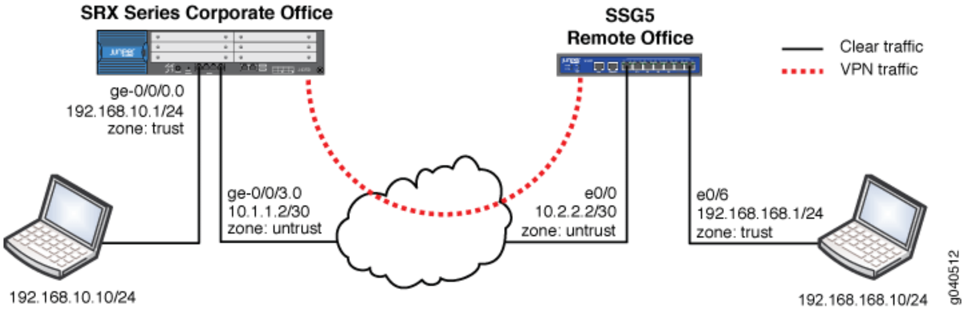
Before you begin:

- Ensure that the remote VPN peer is a Juniper Networks SSG5 Secure Services Gateway device (most commonly used for branch offices).
- Ensure that the internal LAN interface of the device 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 labdomain.com for both VPN peers.

Overview

Figure 5 on page 40 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 5: Network Topology Diagram



NOTE: 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 1.1.1.1. Thus the traffic is outgoing on interface ge-0/0/3.0. Any tunnel policy must consider incoming and outgoing interfaces.

NOTE: 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.

NOTE: 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.

NOTE: 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.

NOTE: 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.
- **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

  NOTE: 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.

NOTE: 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 file name 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@labdomain.com).

NOTE: Currently the Junos OS supports only the RSA algorithm and does not support the Digital Signature Algorithm (DSA). 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 512, 1024, or 2048 bits.

NOTE:

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.

NOTE:
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

- PKI Basic Configuration on page 43
- Configuring a CA Profile on page 44
- Generating a Public-Private Key Pair on page 45
- Enrolling a Local Certificate on page 45
- Loading CA and Local Certificates on page 46
- Configuring the IPsec VPN with the Certificates on page 48

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*.

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 10.10.10.1/24
user@host# set ge-0/0/3 unit 0 family inet address 1.1.1.2/30
```

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 1.1.1.1
```

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 130.126.24.24 offset -0.220645
 sec
```

5. Set the DNS configuration.

```
[edit]
user@host# set system name-server 4.2.2.1
user@host# set system name-server 4.2.2.2
```

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 labdomain.com
```

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

```
[edit]
user@host# set security pki ca-profile ms-ca revocation-check crl
```

NOTE: You can use the **disable** option to disable the revocation check or select the **crl** option to configure the CRL attributes.

3. 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
```

4. 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://labsrv1.labdomain.com/CertEnroll/LABDOMAIN.crl
```

NOTE: Currently you can configure only one URL. Support for backup URL configuration is not available.

5. 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@labdomain.com
   ```

6. 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.

   ```
   user@host> request security pki generate-certificate-request certificate-id ms-cert subject
   "CN=john doe,CN=1.1.1.2,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
   u0qUivMvgKQVCx8hpx99J3EBTurfWL1pCNlBmZggNogb6MbwESOCAwEAAaAwMC4G
   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)
   ```

NOTE: 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. Generate the PKCS10 certificate request to be sent to the CA.

```
user@host> request security pki generate-certificate-request certificate-id id-name
  subject subject-name (domain-name domain-name | ip-address device-ip | email
  email-id) filename filename
```

3. 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://10.10.10.10/certnew.cer certnew.cer
  /var/tmp//...transferring.file.........crYdEC/100% of 1459 B 5864 kBps
user@host> file copy ftp:// 10.10.10.10/CA-certnew.cer CA-certnew.cer
  /var/tmp//...transferring.file.........UKXUWu/100% of 1049 B 3607 kBps
user@host> file copy ftp:// 10.10.10.10/certcrl.crl certcrl.crl
  /var/tmp//...transferring.file.........wpqnpA/100% of 401 B 1611 kBps
```

NOTE: 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: Sales, Country: US, State:
CA, Locality: Sunnyvale,
Common name: Sales
Subject:
Organization: Example, Organizational unit: sales, 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=TACLAB,CN=TACLABSRV1,CN=CDP,CN=Public%20Key%20Services,CN=Services,
CN=Configuration,DC=tacdomain,DC=com?certificateRevocationList?base?
objectclass=cRLDistributionPoint
http://taclabsrv1.tacdomain.com/CertEnroll/TACLAB.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)
```

NOTE: 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: Sales, Country: US, State:
CA, Locality: Sunnyvale,
Common name: Sales
Subject:
Organization: Example, Organizational unit: Sales, Country: US, State:
CA, Locality: Sunnyvale,
Common name: Sales
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=TACLAB,CN=TACLABSRV1,CN=CDP,CN=Public%20Key%20Services,CN=Services,
CN=Configuration,DC=tacdomain,DC=com?certificateRevocationList?base?
objectclass=cRLDistributionPoint
http://taclabsrv1.tacdomain.com/CertEnroll/TACLAB.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.

Syntax (operational mode):
user@host> **show security pki crl ca-profile *ca-profile* [brief | detail]**

Example:

```
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 = Sales, CN = Sales
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 5 on page 40

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 10.10.10.0/24
   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
```

NOTE: 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

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
 202.2.2.2 UP af4f78bc135e4365 48a35f853ee95d21 Main
```

Meaning The output indicates that:

- The remote peer is 2.2.2.2 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 52.

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 57.
- 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 traceoptions to find the mismatch. For more information, see "Troubleshooting IKE, PKI, and IPsec Issues" on page 57.

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 2.2.2.2, Index 20,
Role: Responder, State: UP
Initiator cookie: af4f78bc135e4365, Responder cookie: 48a35f853ee95d21
Exchange type: Main, Authentication method: RSA-signatures
Local: 1.1.1.2:500, Remote: 2.2.2.2:500
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 2.2.2.2 500 ESP:3des/sha1 bce1c6e0 1676/ unlim - 0
>2 2.2.2.2 500 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.

NOTE: 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: 1.1.1.2, Remote Gateway: 2.2.2.2
Local Identity: ipv4_subnet(any:0,[0..7]=10.10.10.0/24)
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 may 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.

NOTE: 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 10.10.10.0/24 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 may 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 57.

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 traceoptions to investigate which ESP packets are experiencing errors and why. For more information, see "Troubleshooting IKE, PKI, and IPsec Issues" on page 57.

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 10.10.10.10 from ethernet0/6** command.

```
ssg5-> ping 10.10.10.10 from ethernet0/6
```

```
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 10.10.10.10, 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 55.
- 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 traceoptions for troubleshooting the routing-related and policy-related issues.

Troubleshooting IKE, PKI, and IPsec Issues

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 traceoptions feature. Using this feature, you can enable a traceoption flag to write the data from the traceoption to a log file, which may 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 traceoptions.

You can enable the traceoptions feature in configuration mode and commit the configuration to use the traceoptions feature. Similarly to disable traceoptions, you must deactivate traceoptions 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
```

NOTE: 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://10.10.10.10/kmd.log
ftp://10.10.10.10/kmd.log 100% of 35 kB 12 MBps
```

Enabling IKE Traceoptions 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 traceoptions.

NOTE: 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
```

NOTE: If you do not specify file names for the <filename> field, then all IKE traceoptions 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.

NOTE: You must commit your configuration to start the trace.

Enabling PKI Traceoptions to View Messages on IPsec

Problem Enable PKI traceoptions 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 Traceoptions to Troubleshoot IKE Setup Issues with Certificates

Problem Configure the recommended settings for IKE and PKI traceoptions.

NOTE: The IKE and PKI traceoptions 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]=
1.1.1.2) 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]=1.1.1.2)
p1_remote=fqdn(udp:500,[0..15]=ssg5.example.net)
p2_local=ipv4_subnet(any:0,[0..7]=10.10.10.0/24)
p2_remote=ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
```

The sample output indicates:

- **1.1.1.2**—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 51.

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 1.1.1.2:500 (Responder) <-> 2.2.2.2:500 { 011359c9 ddef501d -
2216ed2a bfc50f5f [-
1] / 0x00000000 } IP; Error = No proposal chosen (14)
```

The sample output indicates:

- **1.1.1.2**—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:2.2.2.2 is not
recognized.
Nov 7 12:06:36 Phase-1 [responder] failed with error(Authentication failed) for
local=ipv4(udp:500,[0..3]=1.1.1.2) remote=ipv4(any:0,[0..3]=2.2.2.2)
Nov 7 12:06:36 1.1.1.2:500 (Responder) <-> 2.2.2.2:500 { f725ca38 dad47583 -
dab1ba4c ae26674b [-
1] / 0x00000000 } IP; Error = Authentication failed (24)
```

The sample output indicates:

- **1.1.1.2**—Local address.
- **2.2.2.2**—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]=2.2.2.2)
```

The sample output indicates:

- **1.1.1.2**—Llocal address.
- **2.2.2.2**—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 traceoptions 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]=
1.1.1.2) 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]=2.2.2.2)
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]=1.1.1.2) p1_remote=ipv4(udp:500,[0..3]=2.2.2.2)
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 1.1.1.2:500 (Responder) <-> 2.2.2.2:500 { 41f638eb cc22bbfe -
43fd0e85 b4f619d5 [0]
/ 0xc77fafcf } QM; Error = No proposal chosen (14)
```

The sample output indicates:

- **1.1.1.2**—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 10.10.10.0/24. 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]=
1.1.1.2) remote=fqdn(udp:500,[0..15]=ssg5.example.net)
Nov 7 11:52:14 1.1.1.2:500 (Responder) <-> 2.2.2.2:500 { cd9dff36 4888d398 -
6b0d3933 f0bc8e26 [0]
/ 0x1747248b } QM; Error = No proposal chosen (14)
```

The sample output indicates:

- **1.1.1.2** —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 traceoptions feature helps you to gather more information on the debugging issues than is obtainable from the normal log entries. You can use the traceoptions 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

- *VPN Overview*

- Understanding Certificates and PKI on page 3
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 12

Certificate Revocation

Example: Manually Loading a CRL onto the Device

Supported Platforms J Series, LN Series, SRX Series

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 "Example: Generating a Public-Private Key Pair" on page 27.
2. Generate a certificate request. See "Example: Manually Generating a CSR for the Local Certificate and Sending it to the CA Server" on page 35.
3. Configure a certificate authority (CA) profile. See "Example: Configuring a CA Profile" on page 29.
4. Load your certificate onto the device. See "Example: Loading CA and Local Certificates Manually" on page 36.

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.)

NOTE: 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

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
```

NOTE: 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.

Related Documentation

- Understanding Certificate Revocation Lists on page 15
- Digital Certificates Configuration Overview on page 25
- Example: Verifying Certificate Validity on page 70
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 68
- Deleting a Loaded CRL (CLI Procedure) on page 71
- *Public Key Infrastructure Feature Guide for Security Devices*

Example: Configuring a Certificate Authority Profile with CRL Locations

Supported Platforms J Series, LN Series, SRX Series

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

- Requirements on page 69
- Overview on page 69
- Configuration on page 69
- Verification on page 70

Requirements

Before you begin:

1. Generate a key pair in the device. See “Example: Generating a Public-Private Key Pair” on page 27.
2. Create a CA profile or profiles containing information specific to a CA. See “Example: Configuring a CA Profile” on page 29.
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 35.
4. Load the certificate onto the device. See “Example: Loading CA and Local Certificates Manually” on page 36.
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 67.

Overview

In Phase 1 negotiations, you check the CRL list to see if the certificate that you received during an IKE exchange is still valid. If a CRL did not accompany a CA certificate and is not loaded on the device, Junos OS tries to retrieve the CRL through the LDAP or HTTP CRL location defined within the CA certificate itself. If no URL address is defined in the CA certificate, the device uses the URL of the server that you define for that CA certificate. If you do not define a CRL URL for a particular CA certificate, the device gets the CRL from the URL in the CA profile configuration.

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

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

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.

Related Documentation

- Understanding Certificate Revocation Lists on page 15
- Example: Manually Loading a CRL onto the Device on page 67
- Example: Verifying Certificate Validity on page 70
- Deleting a Loaded CRL (CLI Procedure) on page 71
- Deleting Certificates (CLI Procedure) on page 38
- *Public Key Infrastructure Feature Guide for Security Devices*
- *Public Key Infrastructure Feature Guide for Security Devices*

Example: Verifying Certificate Validity

Supported Platforms J Series, LN Series, SRX Series

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

- Requirements on page 70
- Overview on page 70
- Configuration on page 71
- Verification on page 71

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

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
   ```

NOTE: 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.

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

Related Documentation

- Understanding Certificate Revocation Lists on page 15
- Example: Manually Loading a CRL onto the Device on page 67
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 68
- Deleting a Loaded CRL (CLI Procedure) on page 71

Deleting a Loaded CRL (CLI Procedure)

Supported Platforms J Series, LN Series, SRX Series

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.

Related Documentation

- Understanding Certificate Revocation Lists on page 15
- Example: Manually Loading a CRL onto the Device on page 67

- Example: Verifying Certificate Validity on page 70
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 68
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 13

Self-Signed Certificates

Example: Manually Generating Self-Signed Certificates

Supported Platforms J Series, LN Series, SRX Series

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

Requirements

Before you begin, generate a public private key pair. See "Example: Generating a Public-Private Key Pair" on page 27

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 key-pair of the same certificate-id.

Configuration

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.

Related Documentation

- Understanding Self-Signed Certificates on page 17
- Digital Certificates Configuration Overview on page 25
- Using Automatically Generated Self-Signed Certificates (CLI Procedure) on page 74
- *Public Key Infrastructure Feature Guide for Security Devices*

Using Automatically Generated Self-Signed Certificates (CLI Procedure)

Supported Platforms J Series, LN Series, SRX Series

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

- Understanding Self-Signed Certificates on page 17
- Digital Certificates Configuration Overview on page 25

- Example: Manually Generating Self-Signed Certificates on page 73
- *Public Key Infrastructure Feature Guide for Security Devices*

CHAPTER 14

Certificate Chains

Example: Configuring a Device for Peer Certificate Chain Validation

Supported Platforms LN Series, SRX Series

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

Requirements

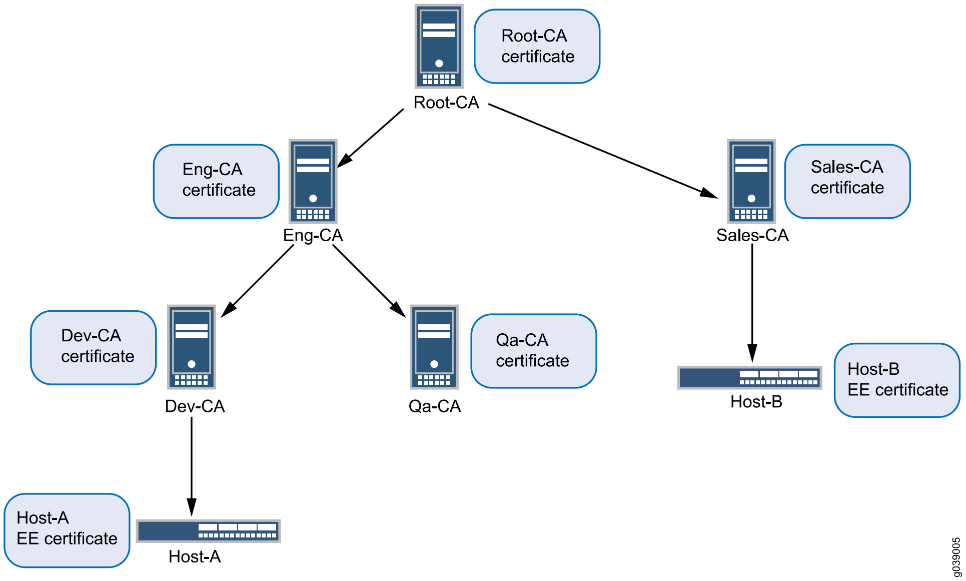
Before you begin, obtain the address of the certificate authority (CA) and the information they require (such as the challenge password) when you submit requests for local certificates.

Overview

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.

This example shows the configuration and operational commands on Host-A, as shown in Figure 6 on page 78. 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 6: Certificate Chain Example



NOTE: 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 *IPsec VPN Feature Guide for Security Devices* for examples of configuring peer devices for VPNs.

Configuration

To configure a device for certificate chains:

- Configure CA Profiles on page 78
- Enroll Certificates on page 80
- Configure IPsec VPN Options on page 81

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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set security pki ca-profile Root-CA ca-identity CA-Root
set security pki ca-profile Root-CA enrollment url http://10.157.88.230:8080/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://10.157.88.230:8080/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://10.157.88.230:8080/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://10.157.88.230:8080/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://10.157.88.230:8080/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://10.157.88.230:8080/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:/;/10.157.88.230:8080/scep/Root/";
  }
  revocation-check {
    crl ;
  }
}
ca-profile Eng-CA {
  ca-identity Eng-CA;
  enrollment {
    url "http:/;/10.157.88.230:8080/scep/Eng/";
  }
  revocation-check {
    crl ;
  }
}
ca-profile Dev-CA {
  ca-identity Dev-CA;
  enrollment {
```

```
      url "http:/;/10.157.88.230:8080/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. 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
```

5. 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)
```

6. 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
```

7. 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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 30.1.1.51
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 30.1.1.31
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 30.1.1.51
   user@host# set external-interface ge-0/0/1.0
   user@host# set local-identity 30.1.1.31
   ```

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 30.1.1.51;
    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

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

- Verifying IKE Phase 1 Status on page 83
- Verifying IPsec Phase 2 Status on page 84

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 UP     285feacb50824495  59fca3f72b64da10  Main           30.1.1.51
```

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   30.1.1.51

      >131073 ESP:3des/sha1 353bacd3 207/  unlim   -   root 500   30.1.1.51
```

IKE and IPsec SA Failure for a Revoked Certificate

- Checking for Revoked Certificates on page 84

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.

Related Documentation

- Understanding Certificate Chains on page 19
- Understanding Certificates and PKI on page 3
- Understanding Certificate Authority Profiles on page 11
- Understanding Certificate Revocation Lists on page 15

CHAPTER 15

Configuration Statements

Security Configuration Statement Hierarchy

Supported Platforms J Series, LN Series, SRX Series

Use the statements in the **security** configuration hierarchy to configure actions, certificates, dynamic virtual private networks (VPNs), firewall authentication, flow, forwarding options, group VPNs, Intrusion Detection Prevention (IDP), Internet Key Exchange (IKE), Internet Protocol Security (IPsec), logging, Network Address Translation (NAT), public key infrastructure (PKI), policies, resource manager, rules, screens, secure shell known hosts, trace options, user identification, Unified Threat Management (UTM), and zones.

Statements that are exclusive to the J Series and SRX Series devices running Junos OS are described in this section.

Each of the following topics lists the statements at a sub-hierarchy of the **[edit security]** hierarchy.

- *[edit security address-book] Hierarchy Level*
- *[edit security alarms] Hierarchy Level*
- *[edit security alg] Hierarchy Level*
- *[edit security analysis] Hierarchy Level*
- *[edit security application-firewall] Hierarchy Level*
- *[edit security application-tracking] Hierarchy Level*
- *[edit security certificates] Hierarchy Level*
- *[edit security datapath-debug] Hierarchy Level*
- *[edit security dynamic-vpn] Hierarchy Level*
- *[edit security firewall-authentication] Hierarchy Level*
- *[edit security flow] Hierarchy Level*
- *[edit security forwarding-options] Hierarchy Level*
- *[edit security forwarding-process] Hierarchy Level*
- *[edit security gprs] Hierarchy Level*
- *[edit security group-vpn] Hierarchy Level*
- *[edit security idp] Hierarchy Level*
- *[edit security ike] Hierarchy Level*
- *[edit security ipsec] Hierarchy Level*
- *[edit security log] Hierarchy Level*
- *[edit security nat] Hierarchy Level*
- [edit security pki] Hierarchy Level on page 89
- *[edit security policies] Hierarchy Level*
- *[edit security resource-manager] Hierarchy Level*
- *[edit security screen] Hierarchy Level*
- *[edit security softwires] Hierarchy Level*
- *[edit security ssh-known-hosts] Hierarchy Level*
- *[edit security traceoptions] Hierarchy Level*
- *[edit security user-identification] Hierarchy Level*

- *[edit security utm] Hierarchy Level*
- *[edit security zones] Hierarchy Level*

Related Documentation

- *Master Administrator for Logical Systems Feature Guide for Security Devices*
- *CLI User Guide*

[edit security pki] Hierarchy Level

Supported Platforms J Series, LN Series, SRX Series

```
security {
  pki {
    auto-re-enrollment {
      certificate-id certificate-id-name {
        ca-profile-name ca-profile-name ;
        challenge-password password ;
        re-enroll-trigger-time-percentage percentage ;
        re-generate-keypair;
      }
    }
    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;
      }
      revocation-check {
        crl {
          disable {
            on-download-failure;
          }
          refresh-interval hours;
          url url-name;
        }
        disable;
        ocsp {
          connection-failure (disable | fallback-crl);
          disable-responder-revocation-check;
          nonce-payload (enable | disable);
          url ocsp-url;
        }
        use-ocsp;
      }
      routing-instance routing-instance-name ;
    }
    traceoptions {
      file filename {
        files number;
        match regular-expression;
```

```
               (no-world-readable | world-readable);
               size maximum-file-size;
            }
            flag flag;
            no-remote-trace;
         }
      }
   }
```

Related Documentation

- Security Configuration Statement Hierarchy on page 87
- *Public Key Infrastructure Feature Guide for Security Devices*
- *Security Policy Applications Feature Guide for Security Devices*

administrator

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
administrator {
   e-mail-address e-mail-address ;
}
```

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify an administrator e-mail address to which the certificate request is sent.

Options e-mail-address *e-mail-address* —E-mail address where the certificate request is sent. By default, there is no preset e-mail address.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

auto-re-enrollment (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
auto-re-enrollment {
   certificate-id certificate-id-name {
      ca-profile-name ca-profile-name ;
      challenge-password password ;
      re-enroll-trigger-time-percentage percentage ;
      re-generate-keypair;
   }
}
```

Hierarchy Level [edit security pki]

Release Information Statement modified in Junos OS Release 9.0.

Description Configure the automatic reenrollment of a local certificate.

Options The remaining statements are explained separately.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

ca-identity (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax ca-identity *ca-identity*;

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement modified in Junos OS Release 11.1.

Description Specify the certificate authority (CA) identity to use in requesting digital certificates.

Options

- ***ca-identity*** —Name of CA identity. This name is typically the domain name of the CA.
- ***routing-instance-name*** —Name of routing instance. The routing instance name is chosen from the list of configured routing instances.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

ca-profile-name

Supported Platforms J Series, LN Series, SRX Series

Syntax ca-profile-name *ca-profile-name*;

Hierarchy Level [edit security pki auto-re-enrollment certificate-id *certificate-id-name*]

Release Information Statement modified in Junos OS Release 9.0.

Description Specify the name of the certificate authority (CA) profile.

Options ***ca-profile-name*** —Name of the specific CA profile.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

ca-profile (Security PKI)

Supported Platforms J Series, LN Series, SRX Series

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;
  }
  revocation-check {
    crl {
      disable {
        on-download-failure;
      }
      refresh-interval hours;
      url url-name;
    }
    disable;
    ocsp {
      connection-failure (disable | fallback-crl);
      disable-responder-revocation-check;
      nonce-payload (enable | disable);
      url ocsp-url;
    }
    use-ocsp;
  }
  routing-instance routing-instance-name ;
}
```

Hierarchy Level [edit security pki]

Release Information Statement modified in Junos OS Release 8.5. Support for **ocsp** and **use-ocsp** options added in Junos OS Release 12.1X46-D20.

Description Configure certificate authority (CA) profile.

Options ***ca-profile-name*** —Name of a trusted CA.

The remaining statements are explained separately.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

certificate-id (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
certificate-id certificate-id-name {
  ca-profile-name ca-profile-name;
  challenge-password password;
  re-enroll-trigger-time-percentage percentage;
  re-generate-keypair;
}
```

Hierarchy Level [edit security pki auto-re-enrollment]

Release Information Statement modified in Junos OS Release 9.0.

Description Specify the certificate authority (CA) certificate to use for automatic reenrollment.

Options ***certificate-id-name*** —Identifier of the certificate that needs automatic reenrollment.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

challenge-password (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax challenge-password *password*;

Hierarchy Level [edit security pki auto-re-enrollment certificate-id *certificate-id-name*]

Release Information Statement modified in Junos OS Release 9.0.

Description 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.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

crl (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
crl {
  disable {
    on-download-failure;
  }
  refresh-interval hours;
  url url-name;
}
```

Hierarchy Level [edit security pki ca-profile *ca-profile-name* revocation-check]

Release Information Statement introduced in Junos OS Release 8.5.

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*—Time interval, in hours, between CRL updates.

  Range — 0 through 8784 hours.
- **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.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

disable (PKI)

Supported Platforms J Series, LN Series, SRX Series

Syntax disable;

Hierarchy Level [edit security pki ca-profile *profile-name* revocation-check]

Release Information Statement modified in Junos OS Release 9.0.

Description Disable revocation checks.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

enrollment (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
enrollment {
  retry number;
  retry-interval seconds;
  url url-name;
}
```

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement introduced in Junos OS Release 9.0.

Description Specify the enrollment parameters for a certificate authority (CA).

Options

- **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, in seconds, between the enrollment retries.

  Range: 0 through 3600

  Default: 900 seconds

- **url** ***url-name*** —Enrollment URL where the Simple Certificate Enrollment Protocol (SCEP) request is sent to the certification authority (CA) as configured in this profile.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

pki

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
pki {
  auto-re-enrollment {
    certificate-id certificate-id-name {
      ca-profile-name ca-profile-name ;
      challenge-password password ;
      re-enroll-trigger-time-percentage percentage ;
      re-generate-keypair;
    }
  }
  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;
    }
    revocation-check {
      crl {
        disable {
          on-download-failure;
        }
        refresh-interval hours;
        url url-name;
      }
      disable;
    }
    routing-instance routing-instance-name;
  }
  traceoptions {
    file {
      filename;
      files number;
      match regular-expression;
      size maximum-file-size;
      (world-readable | no-world-readable);
    }
    flag flag;
    no-remote-trace;
  }
}
```

Hierarchy Level [edit security]

Release Information Statement modified in Junos OS Release 8.5.

Description Configure an IPsec profile to request digital certificates.

Options The remaining statements are explained separately.

Required Privilege Level
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- *Public Key Infrastructure Feature Guide for Security Devices*

pki-local-certificate

Supported Platforms J Series, LN Series, SRX Series

Syntax pki-local-certificate *name*;

Hierarchy Level [edit system services web-management https]

Release Information Statement introduced in Release 9.1 of Junos OS.

Description Specify the name of the certificate that is generated by public key infrastructure (PKI) and authenticated by certificate authority (CA).

Options *name* —Name of certificate.

Required Privilege Level
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation
- *Public Key Infrastructure Feature Guide for Security Devices*

re-enroll-trigger-time-percentage (Security PKI)

Supported Platforms J Series, LN Series, SRX Series

Syntax re-enroll-trigger-time-percentage *percentage*;

Hierarchy Level [edit security pki auto-re-enrollment certificate-id *certificate-id-name*]

Release Information Statement modified in Junos OS Release 9.0.

Description Specify the certificate reenrollment trigger as a 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%, then configure 10 for **re-enroll-trigger-time-percentage**.

Required Privilege Level
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- *Public Key Infrastructure Feature Guide for Security Devices*

re-generate-keypair

Supported Platforms J Series, LN Series, M Series, MX Series, SRX Series, T Series

Syntax

```
<re-generate-keypair>;
```

Hierarchy Level [edit security pki auto-re-enrollment certificate-id]

Release Information Statement introduced in Junos OS Release 8.5.

Description (Optional) Automatically generate a new key pair when auto-reenrolling a router certificate. If this statement is not configured, the current key pair is used.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- Example: Using SCEP to Automatically Renew a Local Certificate on page 34

revocation-check (Security PKI)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
revocation-check {
  crl {
    disable {
      on-download-failure;
    }
    refresh-interval hours;
    url url-name;
  }
  disable;
  ocsp {
    connection-failure (disable | fallback-crl);
    disable-responder-revocation-check;
    nonce-payload (enable | disable);
    url ocsp-url;
  }
  use-ocsp;
}
```

Hierarchy Level [edit security pki ca-profile *ca-profile-name*]

Release Information Statement modified in Junos OS Release 8.5. Support for **ocsp** and **use-ocsp** options added in Junos OS Release 12.1X46-D20.

Description Specify the method the device uses to verify the revocation status of digital certificates.

Options The remaining statements are explained separately.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

routing-instance (Security PKI)

Supported Platforms LN Series, SRX Series

Syntax

```
routing-instance routing-instance-name
```

Hierarchy Level

```
[edit security pki ca-profile ca-profile-name]
```

Release Information Statement modified in Junos OS Release 9.0.

Description Specify the routing-instance to be used.

Options

- ***routing-instance-name***—Name of the routing instance.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

traceoptions (Security PKI)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
traceoptions {
  file {
    filename;
    files number;
    match regular-expression;
    size maximum-file-size;
    (world-readable | no-world-readable);
  }
  flag flag;
  no-remote-trace;
}
```

Hierarchy Level [edit security pki]

Release Information Statement modified in Junos OS Release 8.5.

Description Configure public key infrastructure (PKI) 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**. 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.

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

PART 3

Administration

CHAPTER 16

Operational Commands

clear security pki key-pair (Local Certificate)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
clear security pki key-pair (all | certificate-id certificate-id )
```

Release Information Command introduced in Junos OS Release 8.5.

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

Related Documentation

- show security pki certificate-request (View) on page 129

List of Sample Output clear security pki key-pair all on page 108

Output Fields This command produces no output.

Sample Output

clear security pki key-pair all

```
user@host> clear security pki key-pair all
```

clear security pki local-certificate (Device)

Supported Platforms J Series, LN Series, SRX Series

Syntax clear security pki local-certificate (all | certificate-id ***certificate-id*** | system-generated)

Release Information Command modified in Junos OS Release 9.1.

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.

> NOTE: 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

Related Documentation

- show security pki local-certificate (View) on page 134
- request security pki local-certificate generate-self-signed (Security) on page 121

List of Sample Output clear security pki local-certificate all on page 109
clear security pki local-certificate system-generated on page 109

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
```

request security pki ca-certificate verify (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax **request security pki ca-certificate verify ca-profile** ***ca-profile-name***

Release Information Command introduced in Junos OS Release 8.5.

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

Related Documentation

- ca-profile (Security PKI) on page 93
- show security pki ca-certificate (View) on page 126
- *Public Key Infrastructure Feature Guide for Security Devices*

List of Sample Output request security pki ca-certificate verify ca-profile ca1 (CRL downloaded) on page 110
request security pki ca-certificate verify ca-profile ca1 (CRL not downloaded) on page 110

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

This user has downloaded the certificate revocation list (CRL).

request security pki ca-certificate verify ca-profile ca1 (CRL downloaded)

```
user@host> request security pki ca-certificate verify ca-profile ca1
CA certificate ca1 verified successfully
```

Sample Output

This user has not downloaded the certificate revocation list (CRL).

request security pki ca-certificate verify ca-profile ca1 (CRL not downloaded)

```
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 ca-profile-group load

Supported Platforms J Series, LN Series, SRX Series

Syntax request security pki ca-certificate ca-profile-group load ca-group-name *ca-group-name*
filename *path/filename*

Release Information Command introduced in Junos OS Release 12.1.

Description Manually load a certificate authority (CA) profile group from a specified location.

Options **ca-group-name *ca-group-name***—Load the specified CA group profile.

filename *path/filename*—Directory location and filename of the CA digital certificate.

Required Privilege Level maintenance

Related Documentation

- *show security pki ca-certificate*
- *Public Key Infrastructure Feature Guide for Security Devices*

List of Sample Output request security pki ca-certificate ca-profile-group load on page 111

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

```
user@host> request security pki ca-certificate ca-profile-group load ca-group-name ca-default
filename /var/tmp/firefox-all.pem
Do you want to load this CA certificate ? [yes,no] (no) yes

Loading 196 certificates for group 'ca-default'.
ca-default_1_sysgen: Loading done.
ca-default_2_sysgen: Loading done.
ca-default_3_sysgen: Loading done.
ca-default_4_sysgen: Loading done.
ca-default_5_sysgen: Loading done.
ca-default_6_sysgen: Loading done.
ca-default_7_sysgen: Loading done.
ca-default_8_sysgen: Loading done.
ca-default_9_sysgen: Loading done.
...
...
ca-default_195_sysgen: Loading done.
ca-default_196_sysgen: Loading done.
ca-profile-group 'ca-default' successfully loaded. Success[193] Skipped[3]
```

request security pki ca-certificate enroll (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax request security pki ca-certificate enroll ca-profile *ca-profile-name*

Release Information Command introduced in Junos OS Release 7.5.

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

Related Documentation

- show security pki ca-certificate (View) on page 126
- *Public Key Infrastructure Feature Guide for Security Devices*

List of Sample Output request security pki ca-certificate enroll on page 112

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
```

request security pki ca-certificate load (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax request security pki ca-certificate load ca-profile *ca-profile-name* filename *path/filename*

Release Information Command introduced in Junos OS Release 7.5.

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

Related Documentation

- *show security pki ca-certificate*
- *Public Key Infrastructure Feature Guide for Security Devices*

List of Sample Output request security pki ca-certificate load on page 113

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
```

request security pki crl load (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax request security pki crl load ca-profile *ca-profile-name* filename *path/filename*

Release Information Command introduced in Junos OS Release 8.1.

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

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

List of Sample Output request security pki crl load on page 114

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
```

request security pki generate-certificate-request (Security)

Supported Platforms J Series, LN Series, SRX Series

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>
```

Release Information Command introduced in Junos OS Release 7.5. Support for **digest** option added in Junos OS Release 12.1X45-D10.

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 digest (default).
- **sha256**—SHA-256 digest.

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

Related Documentation

- show security pki certificate-request (View) on page 129

List of Sample Output

request security pki generate-certificate-request on page 116

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+1BPNbCxNWYvyrnSyVYDbFj8oOXyqog8ACDfVL2JBWrPNBYy7imq/K9soDBbAs6
5hZqqwIDAQABoEcwRQYJKoZIhvcNAQkOMTgwNjAOBgNVHQ8BAf8EBAMCB4AwJAYD
VR0RAQH/BBowGIIWdHAxLmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQF
AAOBgQBc2rq1v5SOQXH7LCb/FdqAL8ZM6GoaN5d6cGwq4bB6a7UQFgtoH406gQ3G
3iHOZfz4xMIBpJYuGd1dkqgvcDoH3AgTsLkfn7Wi3x5H2qeQVs9bvL4P5nvEZLND
EIMUHwteolZCiZ70f09Fer9cXWHSQs1UtXtgPqQJy2xIeImLgw==
-----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)
```

request security pki generate-key-pair (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax

request security pki generate-key-pair certificate-id *certificate-id-name*
<size (256 | 384 | 512 | 1024 | 2048 | 4096)>
<type (dsa | ecdsa | rsa)>

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.

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, 512, 1024, 2048, or 4096 bits. Key pair sizes of 256 and 384 bits are compatible with ECDSA. If a key pair size is not specified, the default value, **1024** bits, is applied.

type—The algorithm to be used for encrypting the public/private key pair:

- **ecdsa**—ECDSA encryption with SHA-2 hash
- **dsa**—Digital Signal Algorithm (DSA) encryption with SHA-1 hash
- **rsa**—Rivest Shamir Adleman (RSA) encryption with SHA-1 hash (default)

Required Privilege Level maintenance

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

List of Sample Output request security pki generate-key-pair on page 117

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@switch> request security pki generate-key-pair type rsa size 1024 certificate-id test
Generated key pair test, key size 1024 bits
```

request security pki local-certificate enroll (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
request security pki local-certificate enroll ca-profile ca-profile-name
   certificate-id certificate-id-name challenge-password password  domain-name
   domain-name subject subject-distinguished-name
<email email-address>
<ip-address ip-address>
```

Release Information Command introduced in Junos OS Release 7.5. Serial number (SN) option added to the subject string output field in Junos OS Release 12.1X45.

Description Request that a certificate authority (CA) enroll and install a local digital certificate online by using the Simple Certificate Enrollment Protocol (SCEP).

NOTE: SCEP supports RSA certificates only.

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 16 characters in length.

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.

subject ***subject-distinguished-name***—Distinguished name format that contains the common name, department, company name, state, and country:

- **CN**—Common name
- **OU**—Organizational unit name
- **O**—Organization name
- **ST**—State
- **C**—Country

email ***email-address***—(Optional) E-mail address of the certificate holder.

ip-address ***ip-address***—(Optional) IP address of the router.

Required Privilege Level maintenance

Related Documentation

- show security pki local-certificate (View) on page 134
- *Public Key Infrastructure Feature Guide for Security Devices*

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

```
user@host> request security pki local-certificate enroll 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.
```

request security pki local-certificate export

Supported Platforms J Series, LN Series, SRX Series

Syntax request security pki local-certificate export

Release Information Command introduced in Junos OS Release 12.1.

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

Related Documentation

- *Public Key Infrastructure Feature Guide for Security Devices*

List of Sample Output request security pki local-certificate export on page 120

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
```

request security pki local-certificate generate-self-signed (Security)

Supported Platforms J Series, LN Series, SRX Series

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 ip-address>
```

Release Information Command introduced in Junos OS Release 9.1. Support for **digest** option added in Junos OS Release 12.1X45-D10.

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

email ***email-address***—(Optional) E-mail address of the certificate holder.

Required Privilege Level maintenance and security

Related Documentation

- clear security pki local-certificate (Device) on page 109

- show security pki local-certificate (View) on page 134

List of 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 on page 122

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
```

request security pki local-certificate load

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
request security pki local-certificate load filename ssl_proxy_ca.crt key ssl_proxy_ca.key
   certificate-id certificate id
```

Release Information Command introduced in Junos OS Release 11.4.

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

Required Privilege Level maintenance and security

Related Documentation

- show security pki local-certificate (View) on page 134
- clear security pki local-certificate (Device) on page 109
- request security pki local-certificate verify (Security) on page 124
- *Public Key Infrastructure Feature Guide for Security Devices*

List of Sample Output request security pki local-certificate load on page 123

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
```

request security pki local-certificate verify (Security)

Supported Platforms J Series, LN Series, SRX Series

Syntax request security pki local-certificate verify certificate-id *certificate-id-name*

Release Information Command introduced in Junos OS Release 8.5.

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

Related Documentation

- request security pki local-certificate load on page 123
- show security pki local-certificate (View) on page 134
- clear security pki local-certificate (Device) on page 109
- *Public Key Infrastructure Feature Guide for Security Devices*

List of Sample Output request security pki local-certificate verify certificate-id bme1 (not downloaded) on page 124
request security pki local-certificate verify certificate bme1 (downloaded) on page 124

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

You receive the following response before the certificate revocation list (CRL) is downloaded:

request security pki local-certificate verify certificate-id bme1 (not 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
```

Sample Output

You receive the following response after the certificate revocation list (CRL) is downloaded:

request security pki local-certificate verify certificate bme1 (downloaded)

```
user@host> request security pki local-certificate verify certificate-id bme1
Local certificate bme1 verification success
```

request security pki verify-integrity-status

Supported Platforms J Series, LN Series, SRX1400, SRX3400, SRX3600, SRX5400, SRX5600, SRX5800

Syntax request security pki verify-integrity-status

Release Information Command introduced in Junos OS Release 11.2.

Description Verify the integrity of public key infrastructure (PKI) files.

Required Privilege Level maintenance

Related Documentation

- [edit security pki] Hierarchy Level on page 89

List of Sample Output request security pki verify-integrity-status on page 125

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
```

show security pki ca-certificate (View)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
show security pki ca-certificate
<brief | detail>
<ca-profile  ca-profile-name >
```

Release Information Command modified in Junos OS Release 8.5. Subject string output field added in Junos OS Release 12.1X44-D10.

Description Display information about the certificate authority (CA) public key infrastructure (PKI) digital certificates configured on the device.

> NOTE: 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

Related Documentation

- ca-profile (Security PKI) on page 93
- *request security pki ca-certificate verify (Security)*

List of Sample Output show security pki ca-certificate ca-profile RootCA brief on page 128
show security pki ca-certificate ca-profile RootCA detail on page 128

Output Fields Table 3 on page 126 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 3: show security pki ca-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.

Table 3: show security pki ca-certificate Output Fields *(continued)*

Field Name	Field Description
Issued to	Device that was issued the digital certificate.
Issued by	Authority that issued the digital certificate.
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**).
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.
Distribution CRL	Distinguished name information and the 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 ca-certificate ca-profile RootCA brief

```
user@host> show security pki ca-certificate ca-profile RootCA brief
Certificate identifier: RootCA
  Issued to: RootCA, Issued by: C = US, O = example, CN = RootCA
  Validity:
    Not before: 05- 3-2012 07:15
    Not after: 05- 2-2017 07:15
  Public key algorithm: rsaEncryption(1024 bits)
```

Sample Output

show security pki ca-certificate ca-profile RootCA detail

```
user@host> show security pki ca-certificate ca-profile RootCA detail
Certificate identifier: RootCA
  Certificate version: 3
  Serial number: 0712dc31
  Issuer:
    Organization: example, Country: US, Common name: RootCA
  Subject:
    Organization: example, Country: US, Common name: RootCA
  Subject string:
    C=US, O=example, CN=RootCA
  Validity:
    Not before: 05- 3-2012 07:15
    Not after: 05- 2-2017 07:15
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:ac:b0:c0:11:ac:0c:34:37:04:97:65:c2:b1
    ae:7e:68:e0:fa:37:23:a1:f0:eb:4d:eb:03:89:c9:d9:0d:34:f3:66
    91:97:8c:e9:9c:d4:b5:55:8d:c1:e2:8b:95:08:9d:29:f8:ab:ac:ff
    ae:af:f7:bc:4b:33:f2:eb:b9:e6:13:6d:18:d7:64:a7:85:78:99:41
    4e:b4:fa:bc:3e:1b:5c:26:25:89:03:af:e9:c6:e9:9e:7b:74:1a:1a
    5b:b4:2a:48:78:57:68:e2:5c:0b:71:71:78:ac:a2:23:5f:ca:d2:4a
    38:4c:35:5a:20:cc:44:39:96:26:20:43:bd:75:fd:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Use for key: CRL signing, Certificate signing, Key encipherment,
  Digital signature
  Fingerprint:
    eb:2a:2a:eb:d3:c7:cb:62:65:2e:6a:76:56:b8:af:88:51:8a:30:c9 (sha1)
    cd:43:ae:a4:b2:11:9e:cf:1a:47:fd:7f:0c:ce:d9:fd (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

show security pki certificate-request (View)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
show security pki certificate-request
<brief | detail>
<certificate-id certificate-id-name >
```

Release Information Command modified in Junos OS Release 8.5.

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

Related Documentation

- clear security pki key-pair (Local Certificate) on page 108

List of Sample Output show security pki certificate-request certificate-id user brief on page 130
show security pki certificate-request certificate-id user detail on page 130

Output Fields Table 4 on page 129 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 4: 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 4: 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 user 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 user detail
    Certificate identifier: user
      Certificate version: 3
      Subject:
        Organization: example, Organizational unit: pepsi, Country: IN,
                        Common name: user
      Alternate subject: 102.168.72.124
      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
```

show security pki crl (View)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
show security pki crl
< brief | detail>
<ca-profile ca-profile-name >
```

Release Information Command modified in Junos OS Release 8.5.

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

Related Documentation

- crl (Security) on page 95

List of Sample Output

show security pki crl ca-profile ca2 on page 132
show security pki crl ca-profile ca2 brief on page 132
show security pki crl ca-profile ca2 detail on page 132

Output Fields Table 5 on page 131 lists the output fields for the **show security pki crl** command. Output fields are listed in the approximate order in which they appear.

Table 5: 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.

Table 5: show security pki crl Output Fields *(continued)*

Field Name	Field Description
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
 = Example, OU = SPG QA, CN = 2000-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, OU = SPG QA, CN = 2000-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-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
    21904e5700000000050a       03-16-2007 23:09
    2191cf7900000000050b       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
```

show security pki local-certificate (View)

Supported Platforms J Series, LN Series, SRX Series

Syntax

```
show security pki local-certificate
< brief | detail >
< certificate-id certificate-id-name >
<system-generated>
```

Release Information Command modified in Junos OS Release 9.1. Subject string output field added in Junos OS Release 12.1X44-D10.

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

Related Documentation

- clear security pki local-certificate (Device) on page 109
- request security pki local-certificate generate-self-signed (Security) on page 121

List of Sample Output

show security pki local-certificate certificate-id hello on page 136
show security pki local-certificate certificate-id hello detail on page 136
show security pki local-certificate system-generated on page 137
show security pki local-certificate system-generated detail on page 137
show security pki local-certificate certificate-id mycert - (local certificate enrolled online using SCEP) on page 137
show security pki local-certificate certificate-id mycert detail - (local certificate enrolled online using SCEP) on page 138

Output Fields Table 6 on page 134 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 6: 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.

Table 6: show security pki local-certificate Output Fields *(continued)*

Field Name	Field Description
Serial number	Unique serial number of the digital certificate.
Issued to	Device that was issued the digital certificate.
Issued by	Authority that issued the digital certificate.
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. • **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.
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.
Distribution CRL	Distinguished name information and URL for the certificate revocation list (**CRL**) server.

Table 6: show security pki local-certificate Output Fields *(continued)*

Field Name	Field Description
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
Certificate identifier: hello
  Issued to: cn1, Issued by: DC = local, DC = demo, CN = device1-ABC-A1-CA
  Validity:
    Not before: 08- 8-2012 17:02
    Not after: 08- 8-2014 17:02
  Public key algorithm: rsaEncryption(1024 bits)
```

Sample Output

show security pki local-certificate certificate-id hello detail

```
user@host> show security pki local-certificate certificate-id hello detail
Certificate identifier: hello
  Certificate version: 3
  Serial number: 61ba9da000000000d72e
  Issuer:
    Common name: device1-ABC-A1-CA,
    Domain component: local, Domain component: demo
  Subject:
    Organization: o1, Organization: o2,
    Organizational unit: ou1, Organizational unit: ou2, Country: US, State: CA,
    Locality: Sunnyvale, Common name: cn1, Common name: cn2,
    Domain component: dc1, Domain component: dc2
  Subject string:
   C=US, DC=dc1, DC=dc2, ST=CA, L=Sunnyvale, O=o1, O=o2, OU=ou1, OU=ou2, CN=cn1,
 CN=cn2
  Alternate subject: "ernie@example.net", ernie.example.net, 10.1.2.3
  Validity:
    Not before: 08- 8-2012 17:02
    Not after: 08- 8-2014 17:02
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:b4:14:01:d5:4f:79:87:d5:bb:e6:5e:c1:14
    97:da:b4:40:ad:1a:77:3e:ec:2e:68:8e:e4:93:a3:fe:7c:0b:58:af
    e1:20:27:82:ca:8d:6f:f0:97:d1:ad:fe:df:6c:cb:3c:b0:4f:cc:dd
    ac:d8:69:3f:3c:59:b5:2a:c6:83:e8:b3:94:5e:0a:2d:cd:e2:b0:15
    3e:97:a7:8a:4e:fb:59:f7:20:4c:ba:a8:80:3e:ba:be:69:ef:2b:32
    e4:1a:1c:24:53:1b:d5:c3:aa:d4:25:73:96:76:ea:49:d4:da:7e:3e
    0c:c6:6b:22:43:cb:04:84:0d:25:33:07:6b:49:41:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    ldap:///CN=device1-ABC-A1-CA,CN=everett-win,CN=CDP,CN=Public%20Key
%20Services,CN=Services,CN=Configuration,DC=demo,DC=local?certificateRevocationList?base?
objectClass=cRLDistributionPoint
    http://everett-win.device1.example.net/CertEnroll/device1-ABC-A1-CA.crl
  Use for key: Key encipherment, Digital signature, 1.3.6.1.5.5.8.2.2,
  1.3.6.1.5.5.8.2.2
  Fingerprint:
    76:a8:5f:65:b4:bf:bd:10:d8:56:82:65:ff:0d:04:3a:a5:e9:41:dd (sha1)
    8f:99:a4:15:98:10:4b:b6:1a:3d:81:13:93:2a:ac:e7 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Sample Output

show security pki local-certificate system-generated

```
user@host> show security pki local-certificate system-generated
Certificate identifier: system-generated
  Issued to: JN10B9390AGB, Issued by: CN = JN10B9390AGB, CN = system generated,
CN = self-signed
  Validity:
    Not before: 10-30-2009 23:02
    Not after: 10-29-2014 23:02
  Public key algorithm: rsaEncryption(1024 bits)
```

Sample Output

show security pki local-certificate system-generated detail

```
user@host> show security pki local-certificate system-generated detail
Certificate identifier: system-generated
  Certificate version: 3
  Serial number: e90d42ebd14ef954b3e48c2eed5b30fb
  Issuer:
    Common name: JN10B9390AGB, Common name: system generated, Common name:
self-signed
  Subject:
    Common name: JN10B9390AGB, Common name: system generated, Common name:
self-signed
  Subject string:
    CN=JN10B9390AGB, CN=system generated, CN=self-signed
  Validity:
    Not before: 10-30-2009 23:02
    Not after: 10-29-2014 23:02
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:cb:c8:3f:e6:d3:e5:ca:9d:dc:2d:e9:ca:c7
    5f:b1:f5:3a:f0:1c:a7:55:43:0f:ef:fd:1c:fe:29:09:d5:37:d0:fa
    d6:ee:bc:b8:3f:58:d4:31:fb:96:4f:4f:cc:a9:1a:8f:2e:1b:50:6f
    2b:88:34:74:b2:6d:ad:94:b5:dd:3d:80:87:56:d0:42:50:4d:ac:d7
    8c:21:06:2d:07:1e:f4:d0:c7:85:2e:25:60:ad:1b:b5:b2:d2:1d:c8
    79:67:8c:56:06:04:75:6e:be:4e:99:b8:07:e6:9a:11:fe:b5:ec:c0
    1e:68:da:47:99:1b:b2:c8:07:ab:cd:6e:fe:c1:fd:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    be:1f:21:13:71:cd:9d:de:7a:41:d7:4c:52:8d:3e:d6:ba:db:75:96 (sha1)
    ba:fc:90:4b:5f:a8:66:a3:b9:64:89:9f:e2:45:b5:84 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

Sample Output

show security pki local-certificate certificate-id mycert - (local certificate enrolled online using SCEP)

```
user@host> show security pki local-certificate certificate-id mycert
Certificate identifier: mycert
  Issued to: user, Issued by: DC = local, DC = demo, CN = device1-ABC-A1-CA
  Validity:
    Not before: 11-15-2012 18:58
    Not after: 11-15-2014 18:58
  Public key algorithm: rsaEncryption(1024 bits)
```

Sample Output

show security pki local-certificate certificate-id mycert detail - (local certificate enrolled online using SCEP)

```
user@host> show security pki local-certificate certificate-id mycert detail
Certificate identifier: mycert
  Certificate version: 3
  Serial number: 1f00b50a000000013ad2
  Issuer:
    Common name: device1-abc1-CA,
    Domain component: local, Domain component: demo
  Subject:
    Organization: example-org, Organizational unit: SSD, Country: US,
    Common name: user, Serial number: SRX240-11152012
  Subject string:
    serialNumber=SRX240-11152012, C=US, O=Example-org, OU=SSD, CN=user
  Alternate subject: "user@example.net", user.example.net, 10.150.1.2
  Validity:
    Not before: 11-15-2012 18:58
    Not after: 11-15-2014 18:58
  Public key algorithm: rsaEncryption(1024 bits)
    30:81:89:02:81:81:00:e3:e5:ae:c0:82:af:db:94:01:2f:56:46:50
    7d:3d:0b:0c:f0:1f:1d:7d:c3:aa:d4:4c:a0:cd:23:8b:3f:47:05:ee
    7b:65:42:a0:dc:c4:ac:a7:b6:a6:9f:5c:ea:d8:22:b0:bf:03:75:09
    be:fa:77:cb:d6:67:19:e6:80:fa:a5:7c:93:af:96:66:9f:cc:45:d5
    eb:ab:c1:f0:32:a6:d9:27:1b:80:bb:57:ec:31:a2:e0:2b:e1:42:c0
    92:8a:9b:ed:a6:d2:ec:7c:84:5a:8a:d9:96:a7:7e:40:c3:80:0e:f4
    d6:a2:5d:78:93:3b:7d:d5:8a:f5:de:fb:bc:0d:6d:02:03:01:00:01
  Signature algorithm: sha1WithRSAEncryption
  Distribution CRL:
    ldap:///CN=device1-ABC-A1-CA,CN=everett-win,CN=CDP,CN=Public%20Key%20Services,
CN=Services,CN=Configuration,DC=demo,DC=local?certificateRevocationList?
base?objectClass=cRLDistributionPoint
    http://abc1.device1.example.net/CertEnroll/device1-ABC-A1-CA.crl
  Use for key: Key encipherment, Digital signature, 1.3.6.1.5.5.8.2.2,
  1.3.6.1.5.5.8.2.2
  Fingerprint:
    1f:2f:a9:22:a8:d5:a9:36:cc:c4:bd:81:59:9d:9c:58:bb:40:15:72 (sha1)
    51:27:e4:d5:29:90:f7:85:9e:67:84:a1:75:d1:5b:16 (md5)
  Auto-re-enrollment:
    Status: Disabled
    Next trigger time: Timer not started
```

PART 4

Index

Index