Secure VPN

Viasat Secure VPN User Guide

Prepared by:
Viasat, Inc.
6155 El Camino Real
Carlsbad, CA 92009-1699
Tel: (760) 476-2200
Fax: (760) 929-3941

Prepared for:
National Information Assurance Partnership (NIAP)

Viasat Document No.: 1398812
Rev. 008

NOTICES

Distribution

Viasat® Proprietary – Information, specifications, and features contained in this document are subject to change without notice and should not be construed as a commitment by Viasat Inc. This document is proprietary to Viasat Inc. and shall be protected by a receiving party in accordance with the terms of its contracts and agreements with Viasat Inc., covering all Viasat products.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Viasat Inc.

Trademark

Viasat® and the Viasat logo are registered trademarks of Viasat Inc. in the United States and other countries. All other trademarks, and registered trademarks, are the property of their respective owners.

Copyright

Documentation

The information, specifications, and features contained in this document are subject to change without notice and should not be construed as a commitment by Viasat Inc.

Viasat Inc. assumes no responsibility for any errors that may appear in this document, nor does it make expressed or implied warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Viasat Inc. shall not be liable for incidental or consequential damages in conjunction with, or arising out of the furnishing, performance, or use of this document and the program material it describes.

Publication Information

Revision	Date Released	Comments
001	23 March 2022	Initial Release
002	8 December 2022	Updates per UL comments, release
003	17 March 2023	Updates per UL comments, release
004	19 July 2023	Updates per UL comments, release
005	6 September 2023	Updates per additional UL comments, release
006	14 November 2023	Updates per NIAP checkout package comments
007	20 November 2023	Additional updates per NIAP checkout package comments
008	15 December 2023	Updates per NIAP checkout package comments

Viasat, Inc.

Corporate Headquarters

6155 El Camino Real
Carlsbad, CA 92009-1699

Phone: *(760) 476-2200*
Fax: *(760) 929-3941*

www.Viasat.com

Table of Contents

List of Tables

List of Figures

1 VIASAT SECURE VPN DEVICE OVERVIEW

Viasat Secure VPN virtual Network Device (vND) is intended to provide bump-in-the-wire IPsec encryption to virtual or physical systems deployed behind the device on the Red (Also Known As (AKA) Plaintext) network. The device supports what is known as Gateway (GW) to GW IPsec encryption. The sources and destinations that send data over the IPsec tunnel provided by the device and its remote peer are not aware of their existence. The GW-to-GW configuration is typically used for inter-site encryption.

The Red network could be a virtual or a physical network that supports delivery of user data to the device for encryption and delivery from the device to a destination after decryption. Note that the flows between data sources/destinations on the Red network and the device are not cryptographically protected by the device. End-to-end encryption, such as application layer TLS sessions between the sources and destinations on the red network and their counterparts behind the peer VPN GW could be used if deemed necessary.

On the Red network, the device also provides a second network interface that is used for device remote management (AKA Remote Management Interface (RMI)). This interface is used by an authorized Security Administrator (SA) to access device monitoring and management services provided by the RMI and described later in this document.

Additionally, through the Red network using its management interface, the device can connect to a remote Syslog server to upload its security audit and alert records for real-time security policy violation detection, off-device storage, analysis, non-repudiation and event investigation.

One of the possible device configurations is depicted in Figure 1-1. In this configuration, the device is deployed on the same physical platform as virtual systems that use its IPsec GW services. Each “Application” shown in a separate Virtual Machine (VM) is an example of a client using device’s IPsec services. One or more Virtual Machines (VMs) could be deployed to host various applications that are either sources, destinations or both for the data being transported over the device-provided IPsec tunnel. The virtual network is configured such that the VMs, other than the Viasat Secure VPN vND VM do not have access to physical network interfaces of the underlying platform hardware. Only the vND VM has access to the network interfaces for interfacing to remote IPsec VPN GW peer(s) and applications behind it that exchange data with the applications positioned behind this vND on the Red network. The red dotted line represents the NIAP certification Target of Evaluation (TOE) boundary.

Hyper-V is the only hypervisor that is supported by the vND at the time of this writing. In addition to the Viasat Secure VPN vND VM, one or more guest VMs could be deployed on the platform as required for a particular end user Use Case. These VMs and the hosted applications may use the vND IPsec services or provide unrelated local processing or data storage capabilities.

Figure 1-2 depicts logical interfaces provided by the vND. Some of the interfaces are considered Trusted Channel (TC) interfaces to external IT entities. The Remote Management Interface is considered a Trusted Path (TP) interface and it is used for SA over-the-network management access to the device.

The NIAP-certified configuration for the vND uses Ethernet 0 (Eth 0) network interface for the device Out of Band (OOB) Red Management, Eth1 for the device Plaintext (PT) interface and Eth2 through Eth7 for the device Ciphertext (CT) interface(s). Note that NIAP evaluation only covered using a single CT interface.

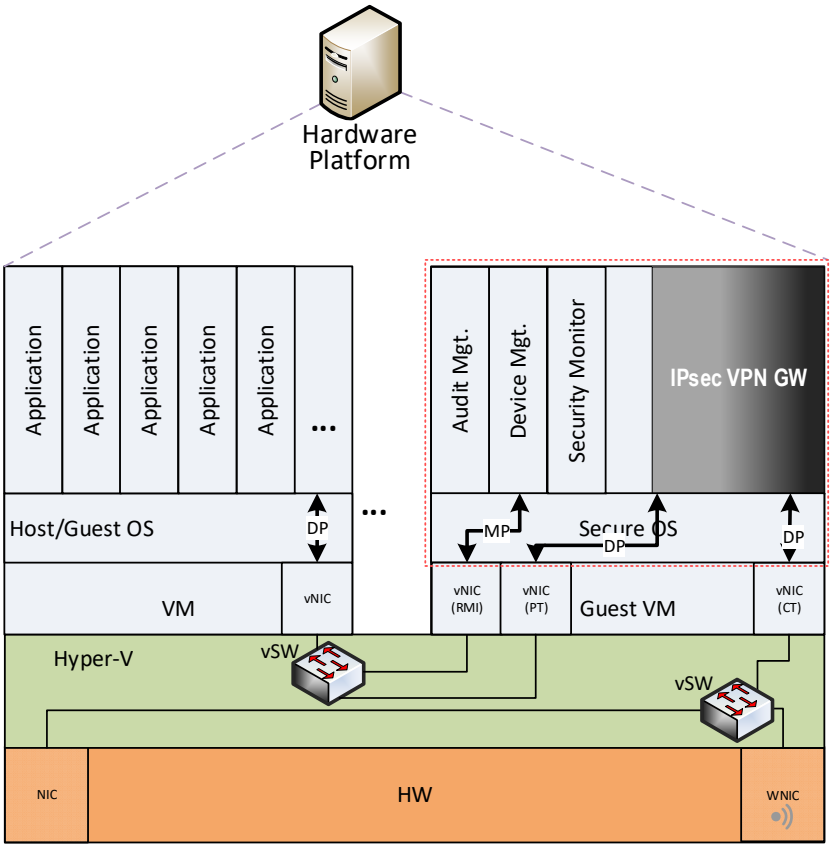


Figure 1-1: An Example vND Deployment

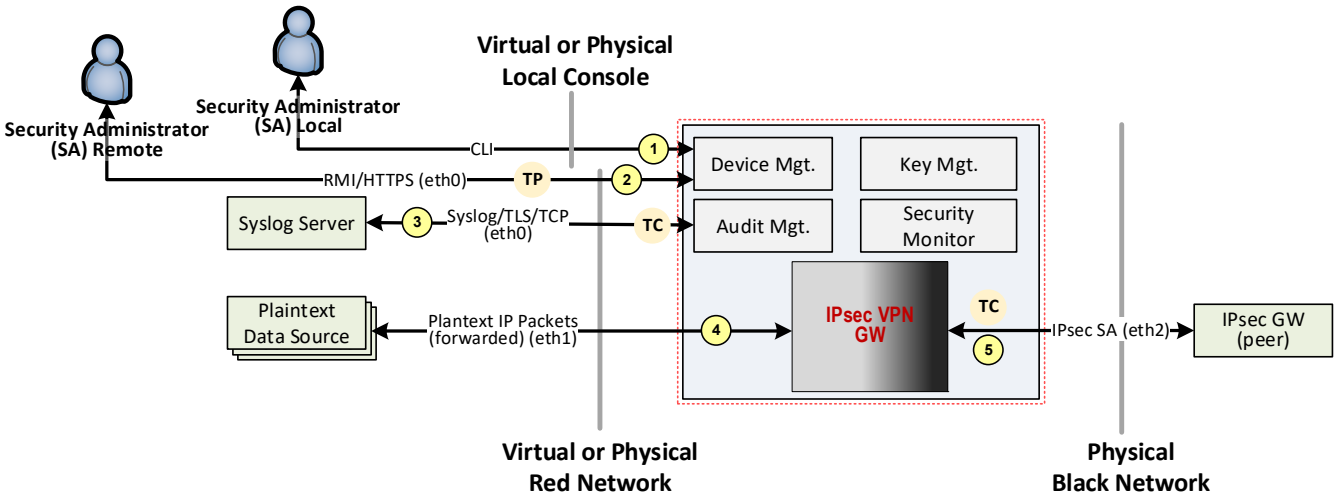


Figure 1-2: vND External Interfaces

2 System Requirements and Device installation

2.1 System Requirements and Device installation

The device is a virtual appliance that runs inside a Microsoft (MS) Hyper-V Type 1 hypervisor Virtual Machine (VM). There are many hardware options that would adequately support this configuration, the hardware requirements largely depend on other VMs that are deployed on the physical server. The NIAP-certified hardware configuration is as follows:

- Server: Dell XPS 8940
  - Processor: 11th Gen Intel Core i5-1140 @ 2.6Ghz
  - RAM: 16Gb Memory
  - Hard drive: 1TB mechanical (spinning) SATA drive
  - Microsoft Windows 10 Pro, version 22H2, includes Hyper-V

Other equivalent hardware platforms may also work; however, they have not been tested by Viasat and are not within the scope of the NIAP-approved device configuration.

Microsoft Windows 10 OS and Hyper-V hypervisor have security configuration guidance provided by Microsoft.

2.2 Meeting Operational Environment (OE) Security Objectives

As a NIAP-certified virtual Network Device (vND), the device must be deployed in an OE that meets the following security objectives. General recommendations for meeting OE requirements are also provided below.

2.2.1 Physical Security (OE.PHYSICAL)

Physical Security (PHYSEC) must prevent unauthorized physical access to the vND. The owner organization must ensure that PHYSEC is adequate for the data hosted and processed by the vND and may include perimeter fencing, gates and lighting, building and equipment room access controls. Guards, video surveillance and intrusion alarms.

2.2.2 vND Only Functions (OE.NO_GENERAL_PURPOSE)

The vND is a dedicated network device that provides IPsec VPN and supporting services. It is not intended to be used for general purpose processing. Extraneous software should not be installed that is not provided by Viasat. The vND only includes the functions required for its operations.

2.2.3 vND Traffic Protection (OE.NO_THRU_TRAFFIC_PROTECTION)

Other than the traffic specifically configured for IPsec protection by the vND, the device does not provide any protection of traffic that traverses it.

2.2.4 Trusted Administrator (OE.TRUSTED_ADMIN)

The owning organization must ensure that the Security Administrators (SAs) authorized to access the vND are trained and trusted to perform their duties. It is expected the SAs will be monitoring vND PKI certificates for expiration and revocation and update such PKI Key Material (KMAT) when required.

2.2.5 Vulnerability Management (OE.UPDATES)

The owning organization must monitor for Viasat vulnerability notifications and update the vND when patches are provided by Viasat. This is intended to remediate any discovered vulnerabilities before they can be exploited by an adversary.

2.2.6 Protection of vND SA Credentials (OE.ADMIN_CREDENTIALS_SECURE)

The owning organization must have credentials management process in place that the SA follows to ensure that vND credentials are protected from unauthorized access. For instance, the vND password is a "something you know" credential and the SA should be prohibited from recording the password and must memorize it.

The password management must follow your organization security policy and procedures. General recommendations are as follows:

1. Password length and composition:
    a. 15 characters with at least one upper case and one lower case letter, at least one digit and at least one special character.
2. Password reuse:
    a. Each new password must be different from the last 10 passwords used.
3. Password aging:
    a. Passwords must be replaced every 60 days.
4. Default manufacturer credentials:
    a. This includes passwords and Key Material (KMAT).
    b. Must be changed prior to deployment.

2.2.7 Protection of Residual Information in vND (OE.RESIDUAL_INFORMATION)

Upon removing the equipment from the system and shelter where it is deployed, the SA must ensure that all sensitive information including KMAT and credentials is removed. Since the vND is deployed as a Virtual Machine (VM) that uses a virtual drive, deleting and zeroizing the VM virtual drive file would remove all vND data. E.g., using Windows cipher.exe tool.

2.2.8 Virtual Machine (VM) Configuration (OE.VM_CONFIGURATION)

The VM for vND must be configured as described in section 2.3. The VM should not have any virtual hardware added, other than described in the section 2.3. Moreover, Virtualization System (VS) features, such as cloning, save/restore, suspend/resume, live migration, etc. are not supported by this vND and should not be used.

2.2.9 Virtual Machine (VM) Networking (OE.CONNECTIONS)

The vND VM virtual networking must be configured as described in sections 2.3 and 7. No Additional virtual network connections or interfaces should be configured.

2.3 Virtual Network Device (vND) Environment Configuration

This vND requires Microsoft (MS) Windows 10 with Hyper-V hypervisor to operate. These Operational Environment (OE) components have been NIAP-certified and must be configured per their NIAP-evaluated Operational and Administrative Guidance[1].

The Hyper-V Virtual Machine (VM) that hosts the vND must be configured to provide OE that is consistent with this deice NIAP-evaluated configuration and is we follows[1]:

- CPU: 1 vCPU (minimum)
- RAM: 3042MB (minimum)
- Dynamic Memory: Enabled
- Hyper-V VM Type: Generation 1 VM
- Virtual Networking: the following interfaces must be configured in this order:
    - Management Interface, used to access the RestAPI, Syslog, and CRL downloads.
        - Could be configured as internal, external or private, depending on how this is used in a system.
        - Mapped to Eth0 in the VM
    - Plaintext Interface
        - Could be configured as internal, external or private, depending on how this is used in a system.
        - Mapped to Eth1 in the VM
    - Ciphertext Interface(s): the device supports up to six ciphertext interfaces. Only one can be used at a time.
        - Could be configured as internal, external or private, depending on how this is used in a system.
        - Mapped to Eth2-7 in the VM
- Hyper-V Integration Services:
    - Time Synchronization (with the Host OS):
        - If Hyper-V VM time synchronization service is desired, select this feature. See section 8.9 for more information on time management.
        - Otherwise, uncheck this feature.
    - Uncheck all other items in this section also

2.4 Initial vND Installation

The initial installation is performed on a platform that has not had the vND installed previously or the vND has been removed. For updating to a new version of a previously installed vND that is operating normally, refer to section 8.8.

The initial installation package is available at Viasat customer portal:

- New users: https://myviasat.force.com/Support/apex/CPAccessRequest
- Existing users: https://myviasat.force.com/Support

[1] Refer to guidance in https://www.niap-ccevs.org/Product/CompliantCC.cfm?CCID=2021.1397 and https://www.niap-ccevs.org/Product/CompliantCC.cfm?CCID=2021.1397

Follow the instructions in the portal to access your purchased Viasat products and download the initial vND software package for "Viasat Secure VPN".

The initial image is not digitally signed, however, ensuring that the download is from the Viasat official portal over HTTPS/TLS ensures that the image is not modified in transit. In addition, the portal also provides an image hash value that can be verified by the user after download to ensure that the image has not been altered.

The following items must be downloaded from the portal:

1. Device initial installation image, version 1.1.7 (vND build number will change as minor updates are posted).
2. This User's Guide, document number 1398812 (latest version from the portal).
3. REST API Reference (as Appendix A to this document), document number 1398812 (latest version from the portal).

The initial vND software distribution is in the form of a Hyper-V VM virtual disk image. Initial installation is as follows:

- On a platform that supports Hyper-V, ensure that is it enabled and operational.
- In Hyper-V, create a VM with the parameters described in section 2.3.
- Create a local directory in Windows where the vND virtual drive will be located. E.g., "c:\virtual_machines\viasat_vpn".
- Copy the Viasat-provided virtual disk image into the above directory.
- Update the Hyper-V configuration to point to the virtual disk image above.
- In Hyper-V management console, select the VM created above and select "Connect" and then "Start". Refer to Hyper-V Operational and Administrative Guidance[2] for details.

Upon startup of the virtual machine the vND is launched and will provide a Command Line Interface (CLI) access for a Security Administrator (SA) access. vND CLI access is described in section 4.1.

2.5 Initial vND Configuration

Upon installation, the device must be configured to be in a secure and operational state, in alignment with its NIAP-certified configuration. The steps to configure the device to this initial level are summarized below and each individual configuration is detailed later in this document.

Prior to using the device for operations, it is recommended to perform the following configuration steps:

1. Login to the CLI as described in section 4.1.
2. Configure Remote Management Interface (RMI) as described in section 4.1
3. Configure device time, as described in section 8.9
4. Configure Security Administrator login password and login banner as described in sections 8.4 and 8.2.
5. Configure PlainText and CipherText interface settings, as described in sections 7.3 and 7.4.

[2] https://www.niap-ccevs.org/MMO/Product/st_vid11087-agd.pdf

6. Configure Domain Name Service (DNS) settings as described in section 7.5.
7. Provision device and Certificate Authority (CA) trust store Key Material (KMAT):
    a. Syslog client KMAT and associated CA trust chain, as described in sections 6.1.1 and 6.2.
    b. RMI over HTTPS server KMAT and associated CA trust chain, as described in in sections 6.1.1 and 6.2.
    c. IPsec device KMAT and associated CA trust chain, as described in sections 6.1.1 and 6.2.
8. Configure and enable security audit delivery over the Syslog interface as described in section 5.1.
9. Configure IPsec service, including IPsec packet filtering rules, as described in section 7.2.

***NOTES**: Device CLI and RMI prompts provide basic guidance on resolving configuration errors. For further guidance, contact Viasat customer care:*
Viasat Customer Care Center:

- *Phone: (888) 272-7232 or (760) 476-2600.*
- *Email: securitysupport@viasat.com*

The device includes an embedded cryptographic module that is not configurable separately from the device management described in this document.
The cryptographic module must not be replaced with any other cryptographic module. This would be an unsupported configuration and is outside of the NIAP certified configuration.

3 SUPPORT

Viasat provides sales and technical support for Viasat Secure VPN product customers. The support can be obtained by contacting Viasat Customer Care Center:

- Phone: (888) 272-7232 or (760) 476-2600
- Email: securitysupport@viasat.com

4 DEVICE MANAGEMENT

4.1 Operator Management Access

The device supports two methods for Security Administrator (SA) device monitoring and management:

- Local Management Interface, AKA Command Line Interface (CLI)
- Remote Management Interface (RMI)
    - The RMI provides REST API services used for managing the vND

The device is deployed as a virtual appliance into Microsoft Hyper-V hypervisor environment as described in the NIAP Security Target (ST). The CLI is accessible through the hypervisor, which emulates a dedicated local console interface. The RMI is accessible from the virtual network interface designated for Red (AKA Plaintext) Management. It can be accessed from the local network where the Management interface is provisioned. Note that a client, such as a Web browser or another client accessing the RMI must be on the same local network (same subnet) as the Management Interface. Routing between a client and the device's Management interface from a different network or a remote subnet is not supported.

The CLI requires a username and password-based login for SA access. The default device login credentials are:

- Username: admin
- Password: password

It is highly recommended that the default password is changed prior to commissioning of the device. The same credentials are used for CLI and RMI login.

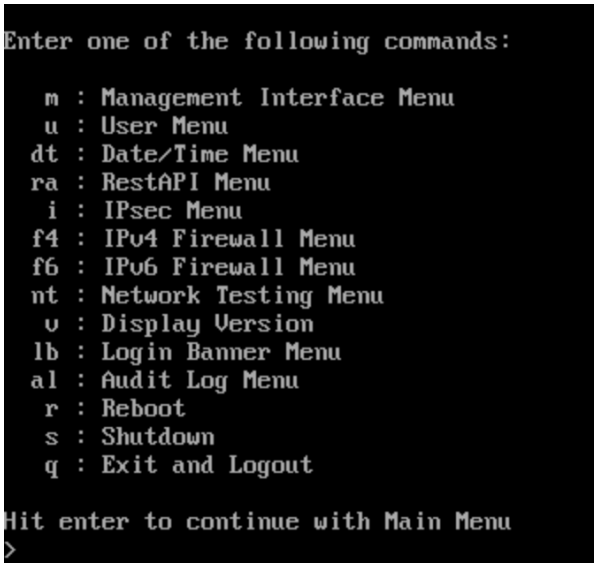
The RMI operates over HTTPS (TLS 1.2), running the REST API interface. An overview of specific RMI commands is provided in the main sections of this guide, for complete REST API reference, including requests, parameters, and corresponding responses, refer to Appendix A. The RMI uses a single factor username and password authentication for SA login. The RMI interface operates over HTTPS/TLS and it comes preconfigured with a default manufacturer certificate installed. This certificate will not be trusted by commercial HTTPS clients. Prior to device deployment, the device owner can make a decision that within the local RMI network Man-in-the-Middle (MITM) threats are not a concern and keep using the default manufacturer certificate. Otherwise, the device owner can install a commercial or system-specific PKI certificate for the RMI. Refer to the section 6.1 for details on device Key Management (KM), including how to replace the default HTTPS certificates and the corresponding private key. The RMI is only accessible over HTTPS protocol on the default port 443. HTTP access over port 80 is not provided by the device.

Red Management IP address must be configured from the CLI prior to accessing the RMI as described below. Alternatively, the default network configuration could be used for initial access to the RMI, which is as follows:

- RMI Address: 169.254.0.1
- Subnet mask: 255.255.255.252

RMI network interface configuration steps:

1. If not logged in, login to the device as described in section 4.1.
2. The main CLI menu will display.



3. Type “m” to open the Management Interface Menu .

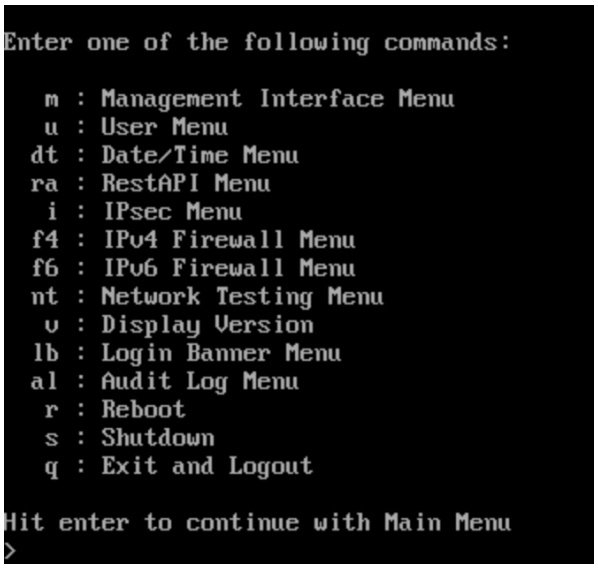
4. Type “sm” then hit Enter, to Show Management Interface IP configuration or “cm” to change the configuration. Selecting “cm”, opens the Change Management Interface IP Address/Netmask menu.

5. Enter the IP address and the netmask. For example: 192.168.1.124/255.255.255.0

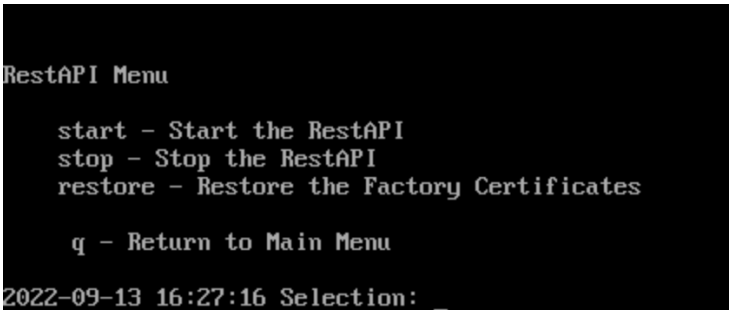
4.1.1 RMI HTTPS Service Control

RMI service is controlled using the CLI. To enable or disable the RMI service, the following sequence is performed:

1. While an SA is logged into the device CLI, from the main menu:



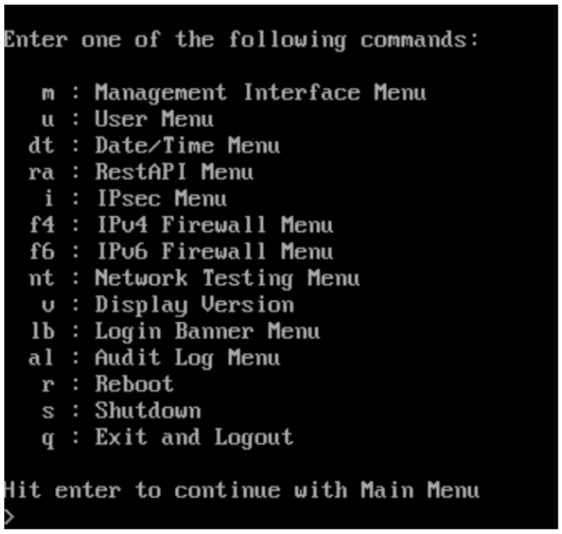
2. Type “ra” to manage the RMI settings and service.
3. RMI (AKA RestAPI) menu is displayed.



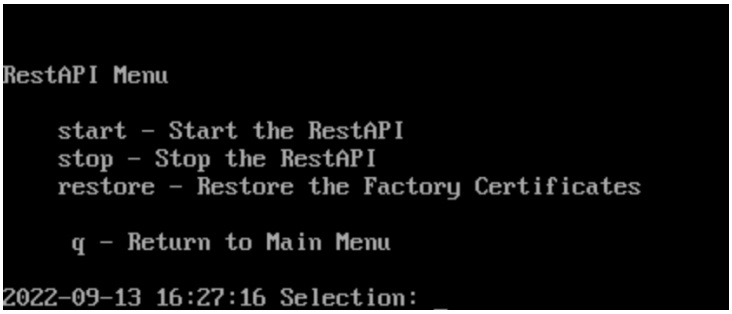
4. Type “start” to start the RMI service or type “stop” to stop the RMI service.
5. The device performs the requested operation.

The RMI uses HTTPS protocol to secure Data in Transit (DIT) between the device and the Security Administrator used workstation/client. Additional details including key management for this interface is provided later in this document. The device comes from the factory with a set of manufacturer device PKI credentials that would not be recognized by any commercial browsers or other tools. It is highly recommended to replace these PKI credentials with your organization approved PKI Key Material (KMAT). In cases where restoring the default PKI credentials is required, the following process is used:

1. While an SA is logged into the device CLI, from the main menu:



2. Type “ra” to manage the RMI settings and service.
3. RMI (AKA RestAPI) menu is displayed.



4. Type “restore” to restore default RMI certificates.
5. The device replaces the Security Administrator installed certificates for the RMI with the default manufacturer keys and certificates, including the entire Certificate Authority (CA) trust chain.

4.1.2 Security Administrator (SA) Login

The CLI and the RMI display a login banner that has been configured by an SA. The CLI displays the banner upon SA login.

For the RMI, the following login sequence must be performed to view the banner and then login into the RMI:

- An SA loads and reads the login banner:
  - **`GET: /login_banner`**
- An SA authenticates to the device and accepts the banner:
  - **`POST: /token`**
    - Parameters: banner_accepted:boolean
    - Body: username:string, password:string

***NOTE**: The banner_accepted must be set to true, otherwise, the login will be denied.*

- The REST API call above returns a session token that must be included in all subsequent REST API requests that require authentication. See the exception below.

No access to the device is allowed prior to a successful completion of the Identification and Authentication (I&A), followed by an authorization check ensuring that the person is in the Security Administrator (SA) role. The only exception is noted below.

1. The login session token that is returned from a successful invocation of the /token REST API, in the “access_token” JSON result must be used for all subsequent REST API calls that require authentication as follows:

- Let the value returned be noted as TOKEN_VALUE
- In consequent HTTP requests, the following HTTP header must be included:
    - Authorization: Bearer TOKEN_VALUE
        - Header name “Authorization”
        - Header value starts with the string “ Bearer ”
        - Header value follows by the TOKEN_VALUE provided by the device earlier

2. The device allows the following functions to be accessible without prior successful Identification, Authentication and Authorization (IAA):

- CLI and RMI:
    - Display login banner

4.1.3 Security Administrator (SA) Login Session

When accessing the device over RMI, upon an SA providing incorrect credentials on multiple successive login attempts, the device will lock the SA account for a configured time period, before the account can be used again. The number of successive failed login attempt is configurable from 3 to 10 by an SA. The delay time-period is also configurable by an SA between 1 and 60 minutes as described in section 8.6.

The CLI does not introduce a login delay or otherwise locks the account upon multiple unsuccessful login attempts. It is assumed that physical access to the device hardware is required to attempt a CLI login, thus brute force password attacks are less likely.

If the RMI SA account is locked due to multiple successive failed login attempts, access can resume after the configured delay time period. The SA can use the CLI at any time to regain management access to the device.

SA login password composition must be as follows:

- Any combination of upper and lower case letters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")"
- Minimum password length is configurable to be between 6 and 20 characters (see section 8.3).

The same Security Administrator (SA) password is used for CLI and RMI SA login. Password change procedure is described in section 8.4.

For the CLI, the SA login credentials being entered are not echoed to the screen, thus protecting them from shoulder surfing type of threats. The CLI and the RMI display a configurable login banner to the SA attempting access.

CLI and RMI logon sessions are subject to a configured inactivity timeout. Upon reaching the timeout, the device terminates the login session and requires the SA to repeat the login process. CLI timeout and RMI timeout values are configured separately in the range from 30 seconds to 120 minutes. The timeout configuration is described in section 8.5.

In addition to the login session idle timeout, a login session can be terminated upon SA request. For the CLI, an SA enters the command "q", which terminates the session. For the REST API, an SA issues the delete request on the login token as follows:

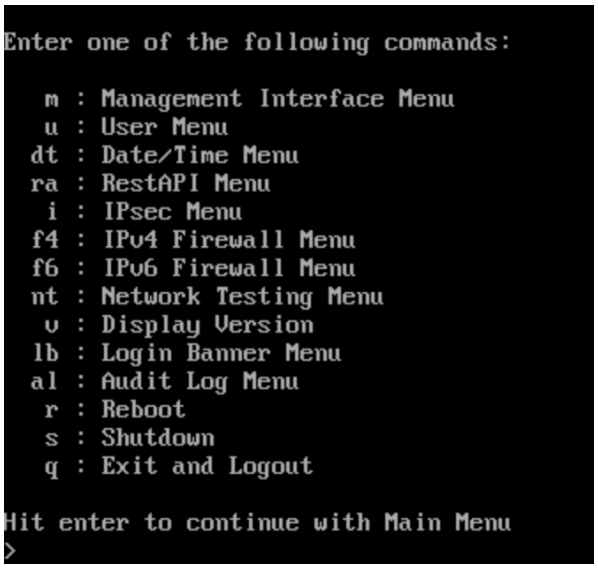
- **`DELETE: /token`**

***NOTE**: Consult your organizational security policy for password composition, aging and reuse. An example of a password composition policy is 15-character password with at least one upper case and one lower case letter, at least one digit and at least one special character.*

4.1.4 CLI Session Termination, Reboot and Shutdown Functions

The CLI provides a function for the Security Administrator (SA) to explicitly terminate the login session. To terminate the session:

1. While an SA is logged into the device CLI, from the main menu:



2. Type "q" to exit the CLI and logout the SA from the device.
3. The device will terminate the CLI session.

For the SA to logout and shut down the device:

1. From the main CLI menu shown above, select the "s" Shutdown option.
2. The device will logout the SA and will shut down.

For the SA to logout and reboot the device:

1. From the main CLI menu shown above, select the "r" Reboot option.
2. The device will logout the SA and will restart.

5 SECURITY AUDIT

Viasat Secure VPN stores security audit and event data locally. When configured, it also delivers security audit and event data over Syslog RFC 5424 protocol to a remote Syslog service. Refer to section 5.1 for details on Syslog client configuration.

The device allocates 10 MB for local storage for audit logs. Upon reaching the storage threshold, the device overwrites the oldest records. If configured by a Security Administrator (SA), security audit records will be delivered to a remote Syslog server. After completing device Syslog service configuration and initiating a connection to the service, the audit log records will be delivered to the remote Syslog service. Note that the records delivered to the remote service are the records generated from the time the device successfully establishes the connection to the time the connection is terminated. Past records, prior to connection establishment and future records after the Syslog service disconnect, will not be delivered, they are only stored locally.

Locally stored audit data can be reviewed by an authorized Security Administrator (SA) role using the Command Line Interface (CLI). The SA role can also delete the entire audit trail if required to free non-volatile audit storage space (see section 5.2).

Upon requesting to delete audit records, the device removes all the internally stored audit records. Note that any records not previously forwarded using the Syslog service prior to SA issuing the audit delete command will be lost.

The device generates audit for the auditable events identified in Table 5-1. Audit record content for all events is identified below and it is augmented for certain events as described in Table 5-1, “Additional Content” column. IPsec GW service logging is configurable using device firewall rules as described in section 7.2.7. Note that when multiple firewall rules, including Allow and Deny match a packet, the first Allow or Deny rule will terminate further packet processing by the firewall. Hence, it is important to insert any Log firewall rules before Allow or Deny rules matching the same packet.

For other auditable events, there are no SA-configurable settings.

Audit record content for all events is as follows:

- Date and time of the event
- Type of event
- Subject identity (if applicable)
- The outcome (success or failure) of the event (if applicable)

Table 5-1: Auditable Events and Additional Record Content

#	Event	Additional Content[3]	Related CC Requirement	Comments
1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).	FIA_AFL.1	Subject identity includes user identification.
2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).	FIA_UIA_EXT.1 FIA_UAU_EXT.2	If subject is a human, subject identity includes user identification.
4	Any attempt to initiate a manual update.	None.	FMT_MOF.1/ManualUpdate	Subject identity includes user identification.
5	All security management activities.	None.	FMT_SMF.1	Subject identity includes user identification.
6	Initiation of update; result of the update attempt (success or failure).	None.	FPT_TUD_EXT.1	Subject identity includes user identification.
7	Discontinuous changes to time.	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).	FPT_STM_EXT.1	Subject identity includes user identification.
8	Failure of Secure Software update	Reason for failure (including identifier of invalid certificate)	FPT_TUD_EXT.2	
9	The termination of a local session by the session locking mechanism.	None.	FTA_SSL_EXT.1 "terminate the session" is selected	Subject identity includes user identification.
10	The termination of a remote session by the session locking mechanism.	None.	FTA_SSL.3	Subject identity includes user identification.
11	The termination of an interactive session.	None.	FTA_SSL.4	Subject identity includes user identification.
12	Trusted Channel (machine to machine) events: -Initiation -Termination -Failure of the trusted	Identification of the initiator and target of failed trusted channels establishment attempt.	FTP_ITC.1 FPT_ITT.1	

[3] The audit content in addition to the content noted earlier in this section

#	Event	Additional Content[3]	Related CC Requirement	Comments
	channel functions.			
13	Trusted Path (operator to device) events: -Initiation -Termination -Failure of the trusted path functions.	None.	FTP_TRP.1/Admin FTP_TRP.1/Join	Subject identity includes user identification.
14	Low storage space for audit events.	None.	FAU_STG_EXT.3/Loc Space	
15	X.509 Certificate Events: -Unsuccessful attempt to validate a certificate -Any addition, replacement, or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store	FIA_X509_EXT.1/ITT FIA_X509_EXT.1/Rev	
16	TLS Server: Failure to establish a TLS session	Reason for failure	FCS_TLSS_EXT.2 FCS_TLSS_EXT.1	
17	HTTPS: Failure to establish a HTTPS session.	Reason for failure	FCS_HTTPS_EXT.1	
18	IPsec: Failure to establish an IPsec SA.	Reason for failure	FCS_IPSEC_EXT.1	
19	TLS Client: Failure to establish a TLS session.	Reason for failure	FCS_TLSC_EXT.1	
21	IPsec: Session Establishment with peer	Entire packet contents of packets transmitted/received during session establishment	FCS_IPSEC_EXT.1	
22	IPsec: Application of rules configured with the 'log' operation	-Source and destination addresses -Source and destination ports	FPF_RUL_EXT.1	Logged if enabled by the SA.

Table 5-2: Auditable Events Examples

#	Event	Example
1	Unsuccessful login attempts limit is met or exceeded.	Event_Type: Lockout, Event_Outcome: Success, Event_Functional_Component: rest_api – for user admin from machine "<IP ADDRESS>" *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Lockout, Event_Outcome: Success, Event_Functional_Component: rest_api – for user admin from machine 1.1.1.1*
2	All use of identification and authentication mechanism.	Event_Type: Login, Event_Outcome: <Failure|Success>, Event_Functional_Component: CLI - for user admin from machine <localhost|IP ADDRESS> *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Login, Event_Outcome: Success, Event_Functional_Component: CLI – for user admin from machine 1.1.1.1*
4	Any attempt to initiate a manual update.	Event_Type: Start software upgrade, Event_Outcome: Success, Event_Functional_Component: *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Start software upgrade, Event_Outcome: Success, Event_Functional_Component:*
5	All security management activities.	Event_Type: Change management IP address, Event_Outcome: Success, Event_Functional_Component: *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Change management IP address, Event_Outcome: Success, Event_Functional_Component:*
6	Initiation of update; result of the update attempt (success or failure).	Failure: • Event_Type: Software Upgrade, Event_Outcome: Failure, Event_Functional_Component: Upgrade file upload requested while upgrade already in progress • Event_Type: Software Upgrade, Event_Outcome: Failure, Event_Functional_Component: Upload file's metadata not found, invalid image type. • Event_Type: Software Upgrade, Event_Outcome: Failure, Event_Functional_Component: Uploaded file's signature could not be verified • Event_Type: Software Upgrade, Event_Outcome: Failure, Event_Functional_Component: Could not validate the uploaded file. • Event_Type: Software Upgrade, Event_Outcome: Failure, Event_Functional_Component: Failure during software upgrade. Success: Event_Type: Software Upgrade, Event_Outcome: Success, Event_Functional_Component: *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Software Upgrade, Event_Outcome: Success, Event_Functional_Component:*
7	Discontinuous changes to time.	Event_Type: Date/Time Change, Event_Outcome: <Success|Failure> , Event_Functional_Component: <CLI|rest_api> - Changed date from <initial date> to <new date> *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Date/Time Change,*

#	Event	Example
		Event_Outcome: Success, Event_Functional_Component: CLI – Changed date from 2023-03-10T21:00:00 to 1970-01-01T00:00:00
8	Failure of Secure Software update	Event_Type: Software Upgrade, Event_Outcome: Failure, Event_Functional_Component: Failure during software upgrade. *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Software Upgrade, Event_Outcome: Failure, Event_Functional_Component: Failure during software upgrade.*
9	The termination of a local session by the session locking mechanism.	Event_Type: Logout, Event_Outcome: Success, Event_Functional_Component: CLI - for user admin from machine localhost *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Logout, Event_Outcome: Success, Event_Functional_Component: CLI – for user admin from machine localhost*
10	The termination of a remote session by the session locking mechanism.	Event_Type: Logout, Event_Outcome: Success, Event_Functional_Component: rest_api – for user admin from machine <IP ADDRESS> *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Logout, Event_Outcome: Success, Event_Functional_Component: CLI – for user admin from machine 1.1.1.1*
11	The termination of an interactive session.	Event_Type: Logout, Event_Outcome: Success, Event_Functional_Component: CLI - for user admin from machine localhost *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Logout, Event_Outcome: Success, Event_Functional_Component: CLI – for user admin from machine localhost*
12	Trusted Channel (machine to machine) events: -Initiation -Termination -Failure of the trusted channel functions.	Initiation: Event_Type: Initiating IPsec IKE SA, Event_Outcome: Success, Event_Functional_Component: Termination: Event_Type: Deleting IPsec IKE SA, Event_Outcome: Success, Event_Functional_Component: sending DELETE for IKE_SA Failure: • Event_Type: Certificate, Event_Outcome: Failure, Event_Functional_Component: ipsec – loading certificate from <peer> failed • Event_Type: Private Key, Event_Outcome: Failure, Event_Functional_Component: ipsec – loading private key from <peer> failed • Event_Type: IPsec, Event_Outcome: Failure, Event_Functional_Component: - Certificate or CRL signature algorithm not supported • Event_Type: IPsec CRL, Event_Outcome: Failure, Event_Functional_Component: – crl fetching failed *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Initiating IPsec IKE SA, Event_Outcome: Success, Event_Functional_Component:*

#	Event	Example
13	Trusted Path (operator to device) events: -Initiation -Termination -Failure of the trusted path functions.	Initiation: Event_Type: Connection Initiated, Event_Outcome: Success, Event_Functional_Component: rest_api from machine <IP Address> Termination: Event_Type: Connection Terminated, Event_Outcome: Success, Event_Functional_Component: from machine <IP Address> Failure: Event_Type: Connection , Event_Outcome: Failure, Event_Functional_Component: ssl failed due to <reason> and <message> srcIP: <IP Address> dstIP: <IP Address> *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Connection Initiated, Event_Outcome: Success, Event_Functional_Component: rest_api from machine 1.1.1.1*
14	Low storage space for audit events.	Event_Type: Audit logs overwritten, Event_Outcome: Success, Event_Functional_Component: Deleted <NUM> audit log record(s) because there is not enough storage left *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Audit logs overwritten, Event_Outcome: Success, Event_Functional_Component: Deleted 10000 audit log record(s) because there is not enough storage left*
15	X.509 Certificate Events: -Unsuccessful attempt to validate a certificate -Any addition, replacement, or removal of trust anchors in the TOE's trust store	Validation Errors: • Event_Type: rest_api Certificate Import, Event_Outcome: Failure, Event_Functional_Component: The Certificates lifetimes are invalid, meaning either expired or not yet valid. Imported Certificate Subject: <Certificate Subject> • Event_Type: rest_api Certificate Import, Event_Outcome: Failure, Event_Functional_Component: Download a CRL for the certificate whose issuer is missing the CRLSign flag. Imported Certificate Subject: <Certificate Subject> • Event_Type: rest_api Certificate Import, Event_Outcome: Failure, Event_Functional_Component: Certificate with identity <DN> has been revoked. Imported Certificate Subject: <Certificate Subject> • Event_Type: rest_api Certificate Import, Event_Outcome: Failure, Event_Functional_Component: The Certificate is untrusted. Verify the loaded CAs are correct. Imported Certificate Subject: <Certificate Subject> Trust Anchors: • Addition: Event_Type: <Service Name> <root_ca|intermediate_ca> New Import, Event_Outcome: Success, Event_Functional_Component: Certificate Subject: <SUBJECT> • Replacement: Event_Type: <Service Name> <root_ca|intermediate_ca> Overwrite Import, Event_Outcome: Success, Event_Functional_Component: Certificate Subject:

#	Event	Example
		<SUBJECT> • Removal Event_Type: <Service Name> <root_ca|intermediate_ca> Deletion, Event_Outcome: Success, Event_Functional_Component: Certificate Subject: <SUBJECT> *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: rest_api Root CA Deletion, Event_Outcome: Success, Event_Functional_Component: Certificate Subject: CN=Hello World*
16	TLS Server: Failure to establish a TLS session	Event_Type: Connection , Event_Outcome: Failure, Event_Functional_Component: rest_api - ssl failed due to <reason> and <message> srcIP: <IP Address> dstIP: <IP Address> *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Connection Failure, Event_Outcome: Failure, Event_Functional_Component: rest_api - ssl failed due SSL handshake failed on verifying the certificate srcIP: 1.1.1.1 dstIP: 2.2.2.2*
17	HTTPS: Failure to establish a HTTPS session.	Event_Type: Connection , Event_Outcome: Failure, Event_Functional_Component: rest_api - ssl failed due to <reason> and <message> srcIP: <IP Address> dstIP: <IP Address> *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Connection Failure, Event_Outcome: Failure, Event_Functional_Component: rest_api - ssl failed due SSL handshake failed on verifying the certificate srcIP: 1.1.1.1 dstIP: 2.2.2.2*
18	IPsec: Failure to establish an IPsec SA.	• Event_Type: Certificate, Event_Outcome: Failure, Event_Functional_Component: ipsec – loading certificate from <peer> failed • Event_Type: Private Key, Event_Outcome: Failure, Event_Functional_Component: ipsec – loading private key from <peer> failed • Event_Type: IPsec, Event_Outcome: Failure, Event_Functional_Component: - Certificate or CRL signature algorithm not supported • Event_Type: IPsec CRL, Event_Outcome: Failure, Event_Functional_Component: – crl fetching failed *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: IPsec, Event_Outcome: Failure, Event_Functional_Component: Certificate or CRL signature algorithm not supported*
19	TLS Client: Failure to establish a TLS session.	Event_Type: Certificate Validation , Event_Outcome: Failure, Event_Functional_Component: syslog – Certificate validation failed *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Certificate Validation, Event_Outcome: Failure, Event_Functional_Component: syslog – Certificate validation failed*
21	IPsec: Session Establishment with peer	Event_Type: IPsec Connection Established, Event_Outcome: Success, Event_Functional_Component: *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: IPsec Connection Established, Event_Outcome: Success, Event_Functional_Component:*
22	FW: Application of rules configured with	Event_Type: iptables LOG rule match, Event_Outcome: Success, Event_Functional_Component: match <Packet Information>

#	Event	Example
	the 'log' operation	*1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: iptables LOG rule match, Event_Outcome: Success, Event_Functional_Component: match*
23	Device Self-Test	Event_Type: Self Tests Passing, Event_Outcome: Success, Event_Functional_Component: Event_Type: Self Tests, Event_Outcome: Failure, Event_Functional_Component: <Self Test Name> *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Self Tests Passing, Event_Outcome: Success, Event_Functional_Component:* *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Self Tests, Event_Outcome: Failure, Event_Functional_Component: Software Integrity* *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Self Tests, Event_Outcome: Failure, Event_Functional_Component: KAT*
24	Start-up Audit Functions	Indicated by Self-test passing and Boot Complete events, also the Remote Syslog service startup: Event_Type: Self Tests Passing, Event_Outcome: Success, Event_Functional_Component: Event_Type: Boot Complete, Event_Outcome: Success, Event_Functional_Component: Startup Complete Event_Type: Remote syslog started, Event_Outcome: Success, Event_Functional_Component: *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Self Tests Passing, Event_Outcome: Success, Event_Functional_Component:* *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Boot Complete, Event_Outcome: Success, Event_Functional_Component: Startup Complete* *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Remote syslog started, Event_Outcome: Success, Event_Functional_Component:*
25	Shut-down of Audit Functions	Indicated by system shut-down event: Event_Type: Shutting Down, Event_Outcome: Success, Event_Functional_Component: System shutting down *1970-01-01T00:00:00+00:00 VSVPN admin: (AUDIT_LOG_EVENT) Event_Type: Shutting Down, Event_Outcome: Success, Event_Functional_Component: System shutting down*

5.1 Delivery to a Remote Syslog Service

As noted earlier, the device supports delivery of its security audit logs to an external Syslog service over the Syslog RFC 5224 protocol. Specifically, the Transport Control Protocol (TCP) in conjunction with the Transport Layer Security (TLS) are used to ensure secure "Trusted Channel (TC)" between the device acting as a Syslog client and the remote Syslog service, acting as a Syslog server.

The following TLS options, cipher suites are supported:

- TLS Protocol Version: 1.2
- Cipher suites: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

To configure the Syslog interface, the following configuration steps need to be completed:

1. Load a Public Key Infrastructure (PKI) Trust Anchor (TA) certificate and an Intermediate CA certificate for Syslog service
   a. See Trust Store Management section 6.2
2. Generate device PKI keys and load device PKI X.509 certificate for Syslog service
   a. See Key Management section 6.1.1
3. Configure Syslog server network parameters
   a. See below in this section, under 5.1.1.

Once the above configuration has been completed and confirmed by a Security Administrator (SA), the SA starts the Syslog service and the device will attempt to establish a secure network connection to the remote Syslog server. Upon successful connection to the remote Syslog server, the device starts sending logs that are contemporaneously generated while the connection is maintained.

Audit records are stored in the local allocated audit store. It must be noted that the local store is limited to 10MB in size. When the size threshold is reached, the oldest records are deleted in 1MB increments and new audit records are persisted. The device stores audit records in ten files, 1MB in size each. When the 10th file reaches its size, the first file (oldest) in the sequence is deleted, freeing 1MB of storage. The prior 10th file becomes 9th and the new empty 10th file is created and is being immediately used to store contemporary audit events as they're being generated by the device. If the remote Syslog server is not configured when the local audit storage limit has been reached or the remote Syslog server is unreachable, it is possible that some records would be lost. For instance, if the Syslog service is not active and the local store reached its 10MB storage threshold, the 1st (oldest) file is deleted to make 1MB of space available. Since the records in the 1st 1MB file have not been transmitted to a remote Syslog service and are now deleted locally, these records have been lost. Hence, it is important for the SA to ensure the Syslog service has been successfully configured and audit logs are being delivered to prevent possible audit data loss.

Device, upon deleting a local audit file will record that event in the current audit trail, including the number of audit records deleted. An example of such audit record is provided below:

2022-09-08T16:05:22+00:00 VSVPN root: (AUDIT_LOG_EVENT) Event_Type: Audit logs overwritten, Event_Outcome: Success, Event_Functional_Component: Deleted 3268 audit log record(s) because there is not enough storage left

5.1.1 Syslog Service Management

After Key Management (KM) settings have been configured as noted above, the following network settings need to be configured for remote Syslog services:

1. IPv4 Address
2. TCP Port or use the default (6514)

3. Syslog server DNS mapping (if Doman Name is used)

Once the above is configured, the Syslog service can be started by an SA. These management activities are supported through the RMI. The following RMI REST API[4] endpoints support Syslog service configuration, enable/disable and provide service status:

- Remote Server Port Set:
  - **`POST: /syslog/server/port`**
    - Parameters: port:integer
- Remote Server Domain Name and IP Address Set[5]
  - **`POST: /syslog/server/dns_mapping`**
    - Parameters: ip_address:string, hostname:string
- Start or Stop Syslog Connection:
  - **`POST: /syslog/action/{service_action},`** `where service_action is {start|stop}`
- Read Remote IP Address and Hostname:
  - **`GET: /syslog/server/dns_mapping`**
- Read Remote Server Port:
  - **`GET: /syslog/server/port`**
- **`Read`** Service Status:
  - **`GET: /syslog/status`**

The Syslog client in the device verifies that the remote Syslog server identity presented as part of the TLS 1.2 handshake matches the identity configured by an SA. The Syslog client ensures that the Domain Name specified by an SA (see the dns_mapping configuration above) when establishing a connection to the remote server is present in the server's PKI certificate. The Domain Name must be in the certificate's Subject Alternate Name (SAN) field or in Common Name. If the SAN field is present, the CN is ignored. If the Domain Name does not match the SA-specified value, the Syslog client terminates the connection and records an audit event in system audit trail. Note that the device does not perform a DNS service lookup of the server IP address, both the domain name and the IP address of the remote server must be provided by the SA prior to establishing the connection.

In situations where the connection to the remote Syslog service is lost due to the service or network outage, the device will attempt to reestablish the connection every sixty seconds until it succeeds, or the SA stops the Syslog client service.

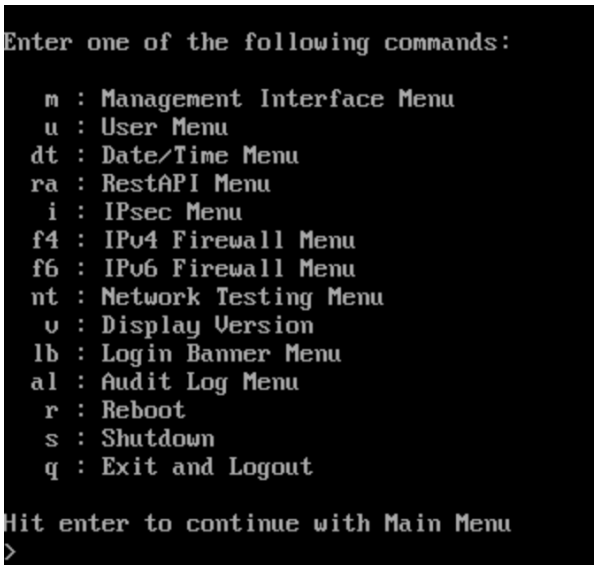
5.2 Audit Management in the CLI

The CLI supports basic security audit management functions, including a basic audit viewer and an option to delete audit logs from the local device audit store.

To view the local logs:

[4] Refer to Appendix A for complete REST API Remote Management Interface specification.
[5] Must be configured correctly or the connection will fail.

1. While an SA is logged into the device CLI, from the main menu:



2. Type “al” to manage device local audit logs.
3. Audit menu is displayed.

4. To delete all local audit logs, enter “d”.
5. Device deletes all local logs.
6. To view local audit logs, enter “v”.
7. The device displays a list of available audit log files (0-9).
8. Enter a desired log file number.
9. The device displays the content of the selected log file.

6 CRYPTOGRAPHIC ALGORITHM AND KEY MANAGEMENT

The device supports multiple keys, algorithms and modes, as described in Table 6-1. General device-supported Public Key Infrastructure (PKI) taxonomy is depicted in Figure 6-1. The device supports up to three levels of the PKI taxonomy. The Intermediate CA is optional, when the intermediate CA is not present, the device supports subscriber (AKA leaf) certificates being issued by the Root CA. The decision on the PKI taxonomy is left to the organization using the device and the device's SA, but it must be within the limits noted in this section.

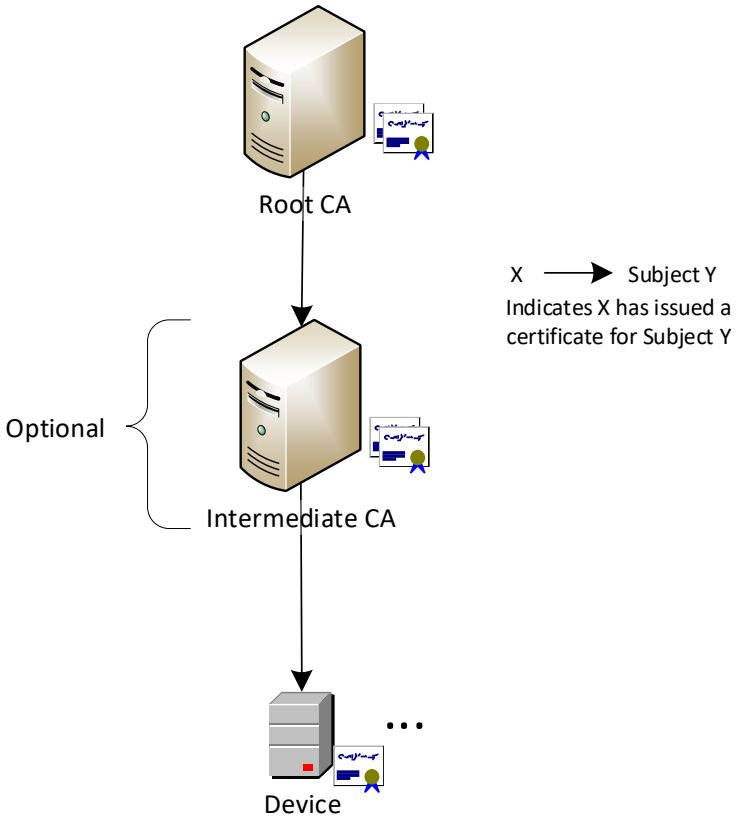


Figure 6-1: Supported PKI Taxonomy

Table 6-1: Interfaces and Cryptographic Key Material (KMAT)

#	Interface	KMAT	KMAT Lifecycle	Comments
1	IPsec VPN	ECDSA Private Key NIST P-256 or P-384 curve, 256-bit or 384-bit respectively	Generated: within crypto module Delivery: not transported Zeroization: by crypto module Data at Rest (DAR): stored internally. Device does not allow access by a user.	This keypair is used for IPsec peer mutual authentication.
2	IPsec VPN	ECDSA Public Key NIST P-256 or P-384 curve, 256-bit or 384-bit respectively	Generated: within crypto module Delivery: transported in CSR and in X.509 certificate Zeroization: by crypto module Data at Rest (DAR): stored internally.	This keypair is used for IPsec peer mutual authentication. There are no confidentially concerns for the public key. Authenticity of the public key is

#	Interface	KMAT	KMAT Lifecycle	Comments
			Device does not allow access by a user.	provided by the X.509 certificate digital signature. Stored with the device's X.509 PKI certificate. The device does not support operator read access to the PKI certificate.
3	Syslog	ECDSA Private Key NIST P-384 curve, 384-bit	Generated: within crypto module Delivery: not transported Zeroization: by crypto module Data at Rest (DAR): stored internally. Device does not allow access by a user.	This keypair is used for TLS mutual authentication.
4	Syslog	ECDSA Public Key NIST P-384 curve, 384-bit	Generated: within crypto module Delivery: transported in CSR and in X.509 certificate Zeroization: by crypto module Data at Rest (DAR): stored internally. Device does not allow access a user.	This keypair is used for TLS peer mutual authentication. There are no confidentially concerns for the public key. Authenticity of the public key is provided by the X.509 certificate digital signature. Stored with the device's X.509 PKI certificate. The device does not support operator read access to the PKI certificate.
5	Rest API (RMI)	ECDSA Private Key NIST P-384 curve, 384-bit	Generated: within crypto module Delivery: not transported Zeroization: by crypto module Data at Rest (DAR): stored internally. Device does not allow access by a user.	This keypair is used for TLS server authentication.
6	Rest API (RMI)	ECDSA Public Key NIST P-384 curve, 384-bit	Generated: within crypto module Delivery: transported in CSR and in X.509 certificate Zeroization: by crypto module Data at Rest (DAR): stored internally. Device does not allow access by a user.	This keypair is used for TLS server authentication. There are no confidentially concerns for the public key. It is distributed freely. Authenticity of the public key is provided by the X.509 certificate digital signature. Stored with the device's X.509 PKI certificate. The device does not support operator read access to the PKI certificate.
7	PlainText (PT) Interface	No encryption or KMAT used		No encryption on this interface.
8	PKI Directory Interface Over HTTP	No encryption or KMAT used		This interface is used for CRL download. CRLs are checked by the device for integrity and authenticity

#	Interface	KMAT	KMAT Lifecycle	Comments
				prior to use. No cryptographic data in transit protection is used on the interface itself.
9	Command Line Interface (CLI)	No encryption or KMAT used		This is a local device management interface. No cryptographic data in transit protection is used on this interface.
10	N/A	Authorized Device Software Source ECDSA Public Key NIST P-384 curve, 384-bit in an X.509 certificate	Delivery: delivered with the device software image. Zeroization: not zeroized, replaced upon software update Data at Rest (DAR): stored internally. Device does not allow access by a user.	This key and the X.509 certificate are used for device software image validation.

As indicated in Table 6-1, the REST API and Syslog interface offer no configurable KMAT options. The IPsec interface supports an SA selecting either NIST P-256 curve or P-384 curve when generating public-private key pair. At the time of device IPsec KMAT enrollment, an SA selects one of the curves for the device to generate the keys and the device allows the SA to download a Certificate Signing Request (CSR) containing the device's public key.

Device signed image delivered for software upgrade contains signer certificate and if used, an intermediate CA certificate. The Root CA certificate used to validate these certificates is embedded in the current image. The device will attempt to connect to CRL Distribution Points (CDPs) specified in the certificates using its management interface. If CDPs do not respond or do not provide the CRLs, the device will proceed with the secure software update process.

In addition to storing its PKI KMAT noted above in its Key Stores, the device also stores PKI CA KMAT in its Trust Stores. Table 6-2 details such KMAT.

Table 6-2: Trust Store Cryptographic Key Material (KMAT)

#	KMAT Owner	KMAT	KMAT Lifecycle	Comments
1	Root CA	ECDSA Public Key NIST P-256[6] or P-384 curve, 256-bit or 384-bit respectively	Delivery: loaded by an SA Zeroization: by crypto module Data at Rest (DAR): stored internally. Device does not allow read access by a user.	This KMAT is loaded and stored in an X.509 certificate. The Root CA certificate (AKA Trust Anchor) must be loaded prior to operation of any Public Key Enabled (PKE) functions that depend on it.

[6] NIST P-256 curve, 256-bit key certificates are supported for IPsec interface only.

#	KMAT Owner	KMAT	KMAT Lifecycle	Comments
2	Intermediate CAs	ECDSA Public Key NIST P-2566 or P-384 curve, 256-bit or 384-bit respectively	Delivery: loaded by an SA Zeroization: by crypto module Data at Rest (DAR): stored internally. Device does not provide access to any device user.	This KMAT is loaded and stored in an X.509 certificate. The Intermediate CA certificate(s) can be loaded prior to operation of Public Key Enabled (PKE) functions. If not loaded prior to operation, the Intermediate CA certificates must be offered by the Remote Entity (RE) during a PKI transaction. E.g., in HTTPS/TLS client Hello message.

NOTE:
Each device interface noted in Table 5 2 uses a separate device certificate and a separate trust store (trust chain), which are configured independently from each other.

The device supported cipher suites for each Trusted Channel (TC) and Trusted Path (TP) interface are detailed in Table 6-3.

Table 6-3: Device Interfaces and Cipher Suites

#	TC, TP Interface	Supported Cipher Suites	Comments
1	IPsec VPN	Authentication: ECDSA, NIST P-256 or P-384 curve, 384-bit and 256-bit keys. IKEv2 SA Key Agreement: Diffie Helman (DH) group 19 and DH group 20 IKEv2 SA symmetric encryption: AES-CBC and AES-GCM, 256-bit key Encapsulated Security Payload (ESP) SA Key Agreement: Diffie Helman (DH) group 19 and DH group 20 ESP symmetric encryption: AES-GCM, 256-bit key Message Integrity: SHA-384 HMAC is used in conjunction with AES-CBC only.	The specific ECDSA curve is based on the device certificate selected at the time of device PKI enrollment. DH group is negotiated with the peer. The DH group 20 is presented by the device as a higher preference option to the peer. The symmetric cipher for IKEv2 is negotiated with the peer. The CBC is presented by the device as a higher preference option to the peer.
2	Syslog	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 NIST P-384 curve	Supports TLS 1.2 protocol per Syslog RFC 5424

#	TC, TP Interface	Supported Cipher Suites	Comments
3	REST API Remote Management Interface	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 NIST P-384 curve	Supports TLS 1.2 protocol
4	PlainText (PT) Interface	No encryption or KMAT used	
5	PKI Directory Interface Over HTTP	No encryption or KMAT used	
6	Command Line Interface (CLI)	No encryption or KMAT used	

It must be noted that each Trusted Channel (TC) and Trusted Path (TP) interface above that utilizes cryptographic Data In Transit (DIT) protection (1-3), uses a separate PKI trust chain from the Root CA down to the device certificate. Each interface chain could be configured to use a different set of CA certificates (trust chain) or the same certificates, but each chain is configured independently from the other chains by an SA.

6.1 Device KMAT Management

6.1.1 CSR Process for PKI Key Generation and Certificate Load

An SA can request the device to generate a PKI key pair, export a Certificate Signing Request (CSR) and import a device certificate that is issued by a Certificate Authority (CA). The process is performed using the RMI as follows:

1. Generate PKI Key Pair and download a CSR:
    a. **`POST: /files/{service}/csr,`** `where service is one of {ipsec | syslog | rest_api}`
        i. Parameters:
            1) ec_curve, one of {P-256 | P-384}
            2) distinguished_name: request certificate DN field value
        ii. If successful, the call returns CSR data (String) in its response.
2. Upload device certificate that was issued by a CA:
    a. **`PUT: /files/{service}/cert,`** `where service is one of {ipsec | syslog | rest_api}`
    b. The certificate must be in PEM format

NOTE:

1. After a CSR is generated, repeated calls to generate another CSR will fail until a certificate is loaded into the device that corresponds to the prior CSR or an SA issues the delete command below for the existing outstanding CSR prior to requesting a new CSR to be generated.

2. When device is requested to generate a PKI key pair and a CSR, the P-256 curve option is only supported for the IPsec interface (service=ipsec)

3. For each TP and TC, the corresponding CA certificate trust store must be provisioned prior to

requesting the device key-pair and CSR generation.
4. This process must be done individually for each TP and TC to provision their KMAT prior to using the interface.
5. When specifying a state as part of the Distinguished Name (DN), using "S=value" will result in an error. Instead, "ST=value" must be used. For example, "…ST=CA…".

When an SA is required to delete an unfulfilled CSR to generate a new key-pair and a new CSR, the following RMI command is used:

- CSR Delete:
  - **DELETE: /files/{service}/csr,** where service is one of {ipsec | syslog | rest_api}

Upon successful conclusion of device certificate load process, the device has a private key stored in its cryptographic module and a corresponding public key in a CA-issued X.509 certificate. The device certificate has been validated, including syntax, composition and revocation status. In addition, the previous device private key and certificate are deleted when the new one has been successfully loaded into the device.

6.1.2 Loading PKI Key Material (KMAT) Generated by a Certificate Authority (CA)

In addition to using the CSR process described above, the device supports loading IPsec PKI KMAT that is generated by a Certificate Authority (CA). This includes device certificate and device private key. This capability is not included in NIAP evaluation.

This process is useful in situations where the CA does not support the CSR process, or the organizational security policy requires the CA to generate public-private key pair.

The device loads CA-generated KMAT in the PKCS #12 format performed from the REST API Remote Management Interface (RMI) as follows:

- **PUT /ipsec/pkcs12_file**
  i. Parameters: password:string
  ii. Body: pkcs12_file:binary

Uploads a PKCS12 file and its password. This device will only upload the issuing CA certificate file from the PKCS12 file. Root CA certificate will not be replaced.

6.1.3 Device KMAT Deletion

Device KMAT deletion is performed from the REST API Remote Management Interface (RMI). All device KMAT can be removed with a single command:

- **DELETE: /files**

This command deletes all device private PKI keys and device certificates. Note that the content of the Trust Stores is not altered.

The following command deletes CA certificates for a specific interface:

- **DELETE: /certificate_authorities/{service},** where service is one of {ipsec | syslog | rest_api}

To delete device's private key, certificate or a CSR, the following command is used:

- **DELETE: /files/{service}/{filename},** where service is one of {ipsec | syslog | rest_api}; filename is one of {key | cert | csr}

The NIAP-certified hardware device configuration uses a rotational hard drive storage device. It is important to note that using a Solid State Drive (SSD) that implements a wear leveling technology may result in remnants of device's KMAT left on the SSD post KMAT deletion.

Upon an authorized SA issuing delete KMAT command, the KMAT is zeroized, then the key file is deleted from the File System (FS). No circumstances other than power loss or a forced Virtual Machine (VM) shutdown during the KMAT deletion would result in remnants of KMAT surviving the delete operation. The SA issuing the delete KMAT command should ensure that the power is maintained and that the VM is not restarted during this operation.

In cases of power loss during the KMAT delete operation, the KMAT or remnants of KMAT may be left in virtual disk storage. After a power loss or forced VM shutdown, repeating the delete operation would remove any remaining KMAT data from the non-volatile storage.

6.2 PKI Trust Store Management

As noted earlier, the device maintains three separate PKI stores for its own certificates and keys as well as for CA certificates, one for each device interface that provides cryptographic DIT protections. Each trust store contains at the minimum a Root CA certificate (AKA Trust Anchor (TA)) and optionally at most one intermediate CA certificate that is signed by the Root CA. Support for different certificate and trust stores for each TC and TP allows using different trust chains for each secure interface. Alternatively, the same certificates could be loaded into each store if desired.

The following command supports loading of a CA certificate into a trust store:

- **`PUT: /certificate_authorities/{service}/{ca_type},`** `where service is one of {ipsec | syslog | rest_api}; ca_type is one of {root_ca | intermediate_ca}`
  - Parameters: ca_file: string

To remove all CA certificates from a trust store, the following command is issued:

- **`DELETE: /certificate_authorities/{service},`** `where service is one of {ipsec | syslog | rest_api}`

NOTES:
Upon loading CA certificates into a device trust store, the device performs certificate chain validation that includes:

- *Certificate syntax, expiration dates and accepted ciphers by this device are correct.*
- *All CA certificates contain the basicConstraints extension with the CA flag set to true.*
- *Root CA certificate is self-signed*
- *Intermediate CA certificate is correctly signed by the Root CA*
- *For each certificate in the path, certificate status validation is performed as described in section 6.3.*

Upon successfully loading CA certificates into one of the device trust stores, the CA certificates that were there provisioned previously (old KMAT) are replaced with the new ones and no longer available for use.

6.3 Certificate Status Validation

The device uses Certificate Revocation Lists (CRLs) to validate X.509 certificate status. The device performs CRL validation of all certificates in the chain at the following times:

- Device's certificates: at the time of new certificate load into the device

- External entity certificate (peer or server): at the time of establishing a Trusted Channel (TC) to the external entity.

The device attempts to download the latest CRL when performing certificate status validation using the CRL Distribution Point (CDP) Universal Resource Identifier (URI) over HTTP protocol per the CDP attribute of each X.509 certificate. If a CRL is not available at the CDP or the certificate does not have a valid CDP HTTP URI, the device will accept the certificate as valid.

As noted in the Domain Name System (DNS) discussion, the device does not support DNS lookups. All Domain Name (DN) to IP address mappings, including for CDP URIs must be configured by a Security Administrator prior to attempting certificate status validation. DNS configuration is described in section 7.5. When a DN to IP address mapping is configured, device firewall rules will also be put in place to allow communication to the configured IP addresses. Should X.509 certificates use CDP URIs that have IP addresses instead of DNs, firewall Allow rules must be configured instead of DNS mapping to allow the device to connect to CDPs and download a CRL.

***NOTES**:*
Device will use X.509 CDP to obtain CRLs over the HTTP protocol.
Domain Name to IP address mapping for all CDPs must be configured prior to use by a Security Administrator
Device management interface is used to download the CRLs, hence each CDP must be reachable through the management interface.
Certificate status validation is different for trusted software update. Refer to section 8.8 for details.

6.4 Cryptographic Algorithm Configuration

The device does not support changes to the key establishment schemes noted in Table 6-3. Where more than one key establishment scheme is available, it is negotiated with the peer per the priority order noted earlier in this document. The device does not support configuration of the order, adding or removing schemes.

For IPsec Trusted Channel (TC), the device only supports Internet Key Exchange (IKE) version 2 (v2). IKEv1 is not supported.

The type of KMAT and cipher suites noted earlier in this document, cannot be altered by an SA. Cryptographic hash algorithms used by the device cannot be altered by an SA. As noted earlier, in cases where multiple cipher suite options exist, they're negotiated with a peer based on the priority specified in this document. One exception is the specification of the Elliptic Curve Cryptography (ECC) curve as the NIST P-256 curve or the NIST P-384 curve upon device key-pair and CSR generation. The NISP P-256 curve option is only supported for the IPsec TC. The specific NIST ECC curve used for IPsec TC is selected based on the device IPsec PKI certificate loaded.

The device implements a Random Number Generator (RNG) that provides random data for cryptographic operations. The RNG operation is compliant with the relevant NIAP Protection Profile (PP) requirements. It is not configurable by an SA.

6.5 PKI Certificate Profile

Several device PKI certificate samples are provided below for Root CA (AKA Trust Anchor (TA)), Intermediate CA and the device. While many fields can be selected by the issuing CA and the device owner, certain fields noted below are mandatory with specific expected values. Certificates that either do not have the noted fields or have the values that deviate from the expected will be rejected by the device.

Table 6-4: Root CA Certificate Example Profile

Field	Sample Values (Mandatory Values Highlighted in Red)
Version	**V3**
Subject	C=US O=Viasat, Inc. CN=Root Certificate Authority Commercial 01
Issuer	Same as the subject (**must be self-signed**)
Signature Algorithm	**Sha384ECDSA or Sha256ECDSA**[8]
Validity Period	10 Years[7]
Public Key Parameters	**ECDSA_P384 or ECDSA_P256**[8]
Public Key	**ECC (384 bits) or ECC (256 bits)**[8]
Key Usage (extension)	Certificate Signing, CRL Signing (06)
basicConstraints (extension)	**Subject Type=CA**, Path Length Constraint=None

Table 6-5: Intermediate CA Certificate Example Profile

Field	Sample Values (Mandatory Values Highlighted in Red)
Version	**V3**
Subject	C=US O=Viasat, Inc. CN=Intermediate Certificate Authority Commercial 01
Issuer	Root CA Identity
Signature Algorithm	**Sha384ECDSA or Sha256ECDSA**[8]
Validity Period	5 Years[7]
Public Key Parameters	**ECDSA_P384 or ECDSA_P256**[8]
Public Key	**ECC (384 bits) or ECC (256 bits)**[8]
Key Usage (extension)	Certificate Signing, Off-line CRL Signing, CRL Signing (06)
basicConstraints (extension)	**Subject Type=CA**, Path Length Constraint=None

[7] The date must be valid when checked by the device, otherwise, it will be rejected
[8] This ECC curve is only supported for the IPsec interface

Table 6-6: Device Certificate Example Profile

Field	Sample Values (Mandatory Values Highlighted in Red)
Version	**V3**
Subject	C=US O=Viasat, Inc. CN=Viasat Secure VPN
Issuer	Issuing CA Identity
Signature Algorithm	**Sha384ECDSA or Sha256ECDSA**[8]
Validity Period	2 Years[7]
Public Key Parameters	**ECDSA_P384 or ECDSA_P256**[8]
Public Key	**ECC (384 bits) or ECC (256 bits)**[8]
Key Usage (extension)	Digital Signature, Key Encipherment
Extended Key Usage (extension)[9]	**Client Authentication (1.3.6.1.5.5.7.3.2)**[10] **or** **Server Authentication (1.3.6.1.5.5.7.3.1)**[11]

[9] Not required for IPsec interface
[10] For Syslog interface only
[11] For RMI interface only

7 EXTERNAL INTERFACES

Device external interfaces are depicted in Figure 1-2 and described in detail in section 1 Viasat Secure VPN Device Overview. In summary, the device supports the following external Trusted Channel (TC) and Trusted Path (TP) interfaces:

1. IPsec gateway (GW) to GW tunnel (Trusted Channel)
   a. Provides IPsec user traffic cryptographic Data-in-Transit (DIT) protection.
2. HTTPS SA to device Remote Management Interface (RMI) (Trusted Path)
   a. Provide device management interface over HTTPS/TLS 1.2 for secure remote management of the device by a Security Administrator (SA). The device HTTP server provides one-way TLS for PKI-based device authentication and for cryptographic DIT protection.
3. Syslog client to external Syslog server interface (Trusted Channel)
   a. Remote security audit delivery interface over TCP, TLS 1.2 from the device embedded Syslog client to a remote Syslog server
4. Command Line Interface (CLI), a local management interface

The device also has several other logical interfaces that are either not direct or not protected using cryptographic Data-in-Transit mechanisms due to the reasons noted later in this section:

1. PKI Directory
   a. Supports download of Certificate Revocation List (CRL).
2. Certificate Authority (CA)
   a. An indirect, man-in-the-loop interface to the CA used to send device CSRs and receive device CA-signed X.509 certificates.
   b. This interface is supported through the device provided management functions to download a CSR from the device and upload device certificate to the device.
3. IPsec client Plaintext (PT) interface
   a. Unencrypted (AKA PT) interface used for IPsec clients to send network packets that will be protected with IPsec by the device and for them to receive packets after they are decrypted by the device.

The IPsec GW to GW interface is activated and deactivated by a command from the RMI. Its status can also be obtained using a command from the RMI. IPsec Security Association (SA) is established autonomously by the device, upon receiving traffic on the plaintext interface that requires IPsec protection or upon a remote GW attempting to establish an SA.

The HTTPS RMI interface is always operational and SA accesses it by initiating a browser connection at the IP address and port for RMI. The connection is terminated upon browser terminating the network session, SA issuing a session termination request or session idle timeout.

Syslog client connection is initiated or terminated by an SA from the RMI as described in 5.1.1.

After system startup, the CLI is always accessible by an SA establishing a local console connection.

The interface to the PKI Directory is used when device performs certificate status validation (see section 6.3). Device uses its management interface to connect to a PKI Directory using the HTTP protocol.

Table 7-1: Device Interfaces[12]

#	Interface	Description	Remote Endpoint	Int. Security
1	Security Administrator (SA) Local Console[13]	Security Administrator (SA) local interface for device management.	Console interface client used by an SA	I&A: Username/Password C: None I: None
2	Security Administrator (SA) RMI[13]	SA Remote Management Interface (RMI) for device management.	Browser or a REST API client used by an SA for RMI	I&A: Server: ECDSA PKI Certificate, TLS 1.2 session Client: Username/Password C: AES-GCM I: GCM
3	Syslog Service Interface[14]	Interface for delivery of device audit records to a remote Syslog server.	Syslog server	I&A: Server: ECDSA PKI Certificate, TLS 1.2 session Client: client ECDSA PKI Certificate, TLS 1.2 session C: AES-GCM I: GCM
4	IPsec GW Client PT Interface[15]	Plaintext (PT) interface for IPsec GW clients to send and receive IPsec-protected packets before encryption and after decryption by the GW that fronts the clients.	Clients on the PT network	I&A: None C: None I: None
5	IPsec GW-GW[16]	Black network interface to an IPsec GW Peer	IPsec Peer GW	I&A: IPsec IKE SA I&A C: AES-GCM or AES-CBC I: SHA-384-HMAC or AES-GCM
6	PKI Directory[14]	Download of CRLs from a PKI Directory over HTTP protocol.	PKI Directory Service	I&A: None[17] C: None I: None
7	PKI CA Interface[18]	Delivery of device CSR to the CA and receiving device certificate from the CA. This is a manual interface. The SA downloads the CSR from the device and loads device certificate after obtaining it from the CA.	CA, manual delivery by SA	I&A: None[19] C: None I: None

[12] In the table: C – Confidentiality, I – Integrity, I&A – Identification and Authentication
[13] For this interface, the device is a server
[14] For this interface, the device is a client
[15] This is a bump-in-the-wire interface
[16] This is a peer-to-peer interface
[17] CRLs are validated for cryptographic authenticity and integrity by the device prior to use
[18] RMI is used by an SA to download a CSR and load a device certificate
[19] Device certificates delivered from the CA are validated for cryptographic authenticity and integrity by the device prior to use

7.1 TLS Configuration

The device uses Transport Layer Security (TLS) version 1.2 for its HTTPS and Syslog interfaces. For these interfaces, the TLS cannot be disabled.

The following configuration steps must be taken prior to enabling TLS:

1. Configure device Key Material (KMAT), as described in section 6.1.
2. Configure device Trust Store, as described in section 6.2

For Syslog, service configuration is described in section 6.1.1. For TLS cipher suites available, refer to Table 6-3. The device does not provide further configuration options for the TLS.

7.2 IPsec

The primary security function of the device is to provide IPsec Gateway (GW) function to clients. The virtual Network Device (vND) is used as a bump in the wire encryptor, positioned between the client systems (typically Virtual Machines (VMs)) on the plaintext side and the black transport network on the device's ciphertext side. The client systems are not aware of the existence of the IPsec GW (hence the "bump in the wire" name). The encryption and decryption are ensured by the position of the vND on the virtual and physical network between the clients and the black transport network.

Note that the device supports IPsec traffic protection of IPv4 only. IPv6 packets can be received by the device, but only Drop and Bypass traffic processing rules are supported for such packets.

Upon IPsec peer disconnect (connection loss), the Security Associations (SAs) will be suspended and the device will attempt to reconnect every five to twenty seconds for a period of one hundred and seventy seconds. If the connection is not reestablished after these attempts, the device will tear down the SAs. An IPsec service restart is required to reestablish the SAs with the peer.

7.2.1 IPsec Service Start/Stop

A Security Administrator (SA) must properly configure the IPsec service as described in this section if not configured previously, and then start/restart the IPsec service. After the IPsec service is properly configured and started, it can process user traffic on its plaintext (PT) and ciphertext (CT) interfaces according to the packet processing rules (described later in this document). The IPsec service can be started and stopped using the following command:

Starting and stopping the IPsec service is performed using RMI as follows:

1. **POST:** `/ipsec/action/{service_action}, where service_action is one of {start|stop}`

If the IPSec service fails to start, review IPsec configuration parameters as described in section 7.2 and restart the IPsec service. Should the problem persist, contact Viasat Customer Case Center.

***NOTES**:*

Upon device restart, the IPsec service will be off. An SA must use the above command to restart the service following a restart, even if the service was running prior to the restart.

Following configuration of the IPsec and service start, the device will attempt to establish an IPsec Security Association with the configured peer Gateway (GW) when:

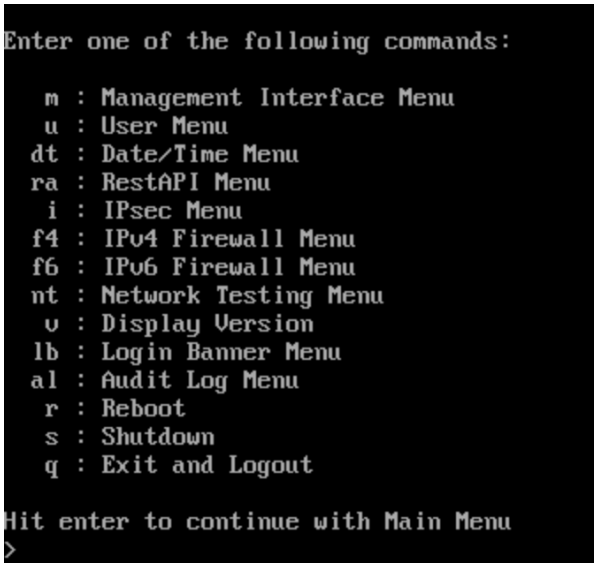
- *Traffic matching the traffic selectors for PROTECT action arrives at the PT interface.*
- *The configured peer initiates SA setup.*

7.2.2 IPsec Security Association Lifetime Configuration

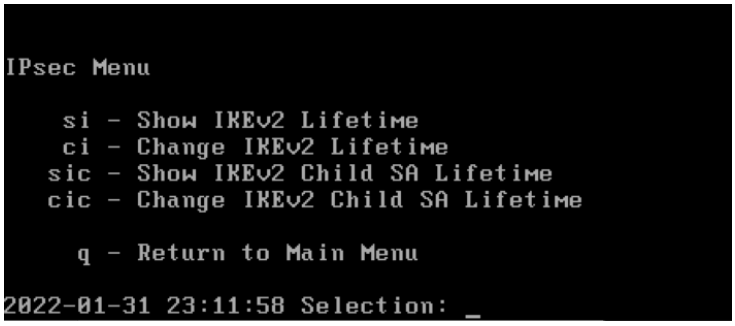
The device supports configuring the IKEv2 Security Association lifetime and the Encapsulated Security Payload (ESP) (AKA Child) Security Association lifetime. Both are configured using the CLI.

To configure the IKEv2 Security Association lifetime, the following steps are required.

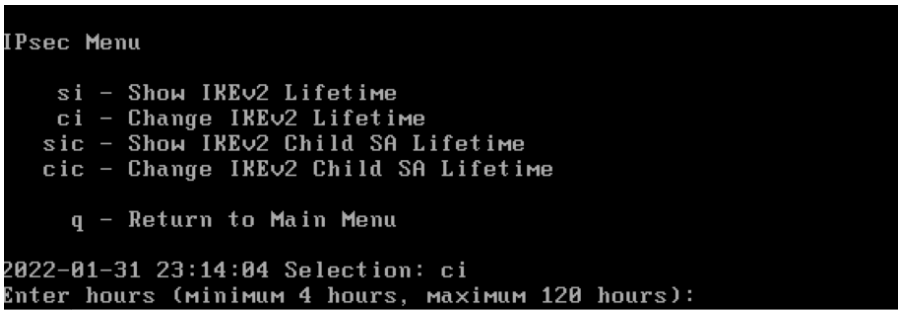
1. If not logged in, login to the device as described in section 4.1.
2. The main CLI menu will display. From the main CLI menu, select “i” to open the IPsec Menu.



3. In the IPsec Menu, select “si” for Show IKEv2 Lifetime to view the set value, or “ci” for Change IKEv2 Lifetime to change the value.

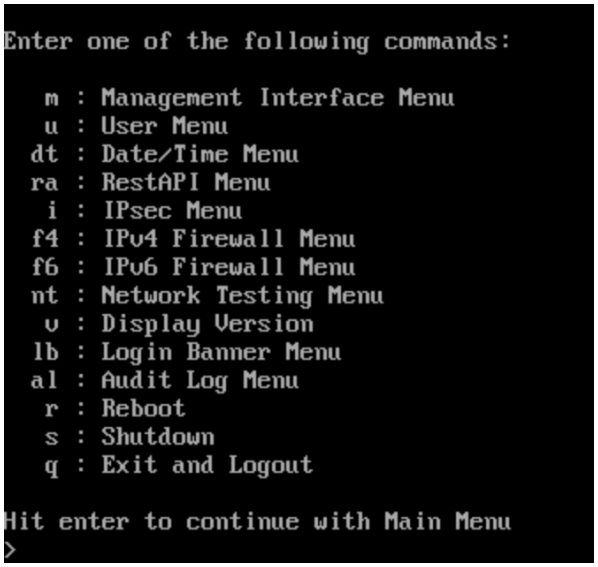


4. Set the desired value within the range of 4 hours (minimum) to 120 hours (maximum) and hit Enter.

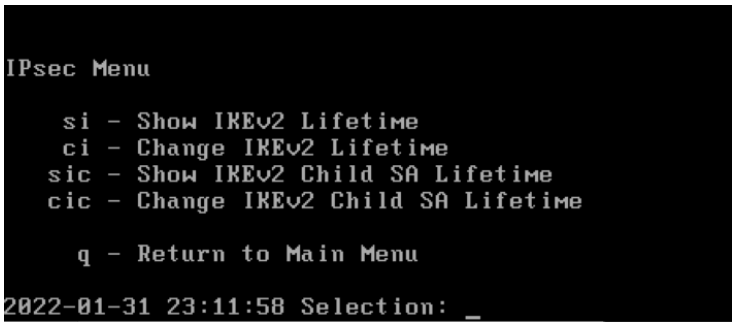


To configure the Child Security Association lifetime, the following steps are required:

1. From the main CLI menu, select “i” to open the IPsec Menu.



2. In the IPsec Menu, select “sic” for Show IKEv2 Child SA Lifetime” to view the set value, or “cic” for Change IKEv2 Child SA Lifetime to change the value.



3. Set the desired value within the range of 2 hours (minimum) to 48 hours (maximum) and hit Enter.

```
IPsec Menu

    si - Show IKEv2 Lifetime
    ci - Change IKEv2 Lifetime
   sic - Show IKEv2 Child SA Lifetime
   cic - Change IKEv2 Child SA Lifetime

     q - Return to Main Menu

2022-01-31 23:16:12 Selection: cic
Enter hours (minimum 2 hours, maximum 48 hours):
```

7.2.3 IPsec Peer Reference Identifier

The device supports configuring peer IPsec Gateway (GW) black IP address. The IP address is used to establish a Security Association (SA) with the peer. In addition, the device requires a Security Administrator (SA) to configure IPsec peer's Distinguished Name (DN). The Configured DN is checked against the peer's X.509 DN field and if it matches, the connection process is allowed to proceed. Otherwise, the connection is terminated, and a security audit record is generated indicating the failure.

Peer's CipherText (CT) (AKA black) IP address is configured using the RMI as follows:

1. Read configured peer CT IP address:
    a. **GET /ipsec/remote_host/ip_address**
2. Set peer CT IP address:
    a. **POST /ipsec/remote_host/ip_address**
        i. Parameters: ip_address:string

The IPsec peer GW DN is configured using the RMI as follows:

2. Read IPsec DN:
    a. GET /ipsec/{endpoint}/dn
        i. Where endpoint is either "local" for the device's DN or "remote" for the expected peer's DN.
3. Configure IPsec peer DN:
    a. POST /ipsec/remote_host/dn
        i. Parameters: remote_dn:string

***NOTES**:*
Device's DN comes from the device IPsec certificate and cannot be manually configured. Peer's DN must be configured by a Security Administrator prior to attempting an IPsec connection.
An example IPsec Peer DN configuration parameter:
CN=VPN GW027, O=Viasat, L=Carlsbad, S=California, C=US
It must be ensured that peer identifier DN is unique. It is assumed that the PKI infrastructure used for the device will ensure DN uniqueness across all PKI subscribers under the same Root CA.

7.2.4 IPsec Device and Peer Port

The devices port used for IPsec Internet Key Exchange (IKE) and the remote peer's port must be configured prior to IPsec service start. To configure the device's and the peer's ports, the following steps are performed through the RMI:

1. Read configured port:

2. **GET /ipsec/{endpoint}/port,** where **endpoint** is one of {local_host|remote_host}

2. Set local or remote port:

   a. **POST /ipsec/{endpoint}/port,** where **endpoint** is one of {local_host|remote_host}

      i. Parameters: port:integer

7.2.5 Packet Filtering

Packet filtering is designed to enforce network traffic flow policy at the IPsec GW Policy Enforcement Point (PEP) level. The device allows an authorized SA to configure packet filtering criteria, actions for a packet matching the criteria and whether packet logging is required. The packet filtering criteria is on the basis of the following:

- IPv4
  - Source address
  - Destination Address
  - Protocol
- IPv6
  - Source Address
  - Destination Address
  - Next Header (Protocol)
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

The actions supported are:

- Allow (protect with IPsec)
- Drop
- Bypass

In addition to enforcing the packet processing rules above, the rules are in the order specified by an SA. The first traffic matching rule is triggered. Hence, a care must be taken where multiple rules may match network packets.

The packet processing rules are configured using the combination of device firewall rules and IPsec Security Policy Database (SPD) rules. Through the use of the RMI IPsec Traffic Selectors endpoint, an SA can configure these rules which will automatically be inserted into the device firewall and IPsec SPD. Note that any packet processing rules created using the IPsec Traffic Selectors endpoint are only in effect while the IPsec service is running. An example of how to use the IPsec Traffic Selectors endpoint is provided in section 7.2.6. An example of using a manual approach configuring device Firewall (FW) directly can be found in section 7.2.7. Since the device does not support IPv6 IPsec protection, the RMI IPsec Traffic Selectors endpoint is not used for configuring the supported Block and Bypass IPv6 packet processing rules. These must be configured directly in the IPv6 firewall.

Packet processing rules that match ingress or egress network packets can generate a security audit log. Enabling or disabling IPsec traffic logging is configurable by a Security Administrator using the FW rules.

To enable or disable packet match event logging, a Security Administrator would use standard Linux iptables commands in conjunction with the CLI "Enter iptables command" menu option described in section 7.2.7. Please refer to the Linux iptables reference below for the specific syntax[20].

***NOTES**:*
IPv6 Traffic Processing rules supported are Block and Bypass only. They must be configured directly in IPv6 Firewall settings. The device does not support IPsec encryption of IPv6 packets.
Packets that do not match any configured rules are dropped by default. This behavior is referred to as "block by default, allow by exception".

A helpful perspective on packet processing rule configuration is from the desired packet action stand point. The desired action (Allow, Drop, Bypass) will determine the configuration steps required to implement it. The Table 7-2 provides a summary for packet processing rule and action configuration approach.

Table 7-2: Packet Processing Rule Configuration

#	Desired Action	Configuration Steps	Comments
1	Drop Specific Packets on Black Interface	Configure a rule on the firewall (FW) to match the packets with the FW REJECT action for the black interface(s).	Note that all packets are dropped by default.
2	Drop Specific Packets on Red Interface	Configure a rule on the firewall (FW) to match the packets with the FW REJECT action on the red interface. OR Configure a Traffic Selector using the RMI to match the packets with the DROP action.	Note that all packets are dropped by default. Traffic selector configuration steps are described in section 7.2.6.
3	Bypass IPsec Encryption from Red to Black	Configure a forward rule on the firewall (FW) to match the packets with the FW ALLOW action coming from the red interface. OR Configure a Traffic Selector using the RMI to match the packets with the BYPASS action.	Note: A Bypass Traffic Selector rule configured using the RMI adds both Red to Black and Black to Red bypass rules. When configuring the FW directly through the CLI, rules must be added for ingress and egress directions separately.
4	Bypass IPsec Encryption from Black to Red	Configure a forward rule on the firewall (FW) to match the packets with the FW ALLOW action going to the red interface. OR Configure a Traffic Selector using the RMI to match the packets with the BYPASS action.	Note: A Bypass Traffic Selector rule configured using the RMI adds both Red to Black and Black to Red bypass rules. When configuring the FW directly through the CLI, rules must be added for ingress and egress directions separately.
5	Allow Specific	Configure a Traffic Selector using the RMI to	Note: An Accept Traffic

[20] There are many resources available that describe iptables command syntax. This is one example: https://man7.org/linux/man-pages/man8/iptables.8.html. Refer to 9.1Appendix B for more details.

#	Desired Action	Configuration Steps	Comments
	Protected Packets on Black Interface	match the packets with the ACCEPT action.	Selector using the RMI adds both Red to Black and Black to Red accept rules.
6	Allow Specific Packets to be Protected from Red Interface	Configure a Traffic Selector using the RMI to match the packets with the ACCEPT action.	Note: An Accept Traffic Selector using the RMI adds both Red to Black and Black to Red accept rules, so an explicit Black to Red Traffic selector is not required.

7.2.6 Configuring IPsec Traffic Policy

When Traffic Policy is configured using this method, appropriate ingress and egress FW rules are automatically provisioned and the IPsec SPD entries are put in place. As noted earlier, the FW policy is only active when the IPsec service is running. When the IPsec service is shut down, any explicit FW rules provisioned directly into the FW are being enforced as well as the default DENY rule for any packets that do not match any explicit rules.

As a simple example of the traffic policy configuration, let's assume the following three traffic selectors are required:

1. Accept all traffic between 1.1.1.0/24 and 2.2.2.0/24 networks.
2. Exclude ICMP traffic, which will be dropped.
3. Bypass any traffic between the 3.3.3.0/24 source port 4500 and 4.4.4.0/24 network destination port 4800.

- Accept Traffic
  - Source Address: 1.1.1.0/24
  - Destination Address: 2.2.2.0/24
- Drop Traffic
  - Source Address: 1.1.1.0/24
  - Destination Address: 2.2.2.0/24
  - Protocol: icmp
- Bypass Traffic
  - Source Address: 3.3.3.0/24
  - Destination Address: 4.4.4.0/24

To configure the IPsec Traffic Selectors using the RMI, the following step is required:

Set the IPsec Traffic Selectors:

1. POST /ipsec/traffic_selectors
   a. Body: JSON Parameter List (protocol, source_port and destination_port parameters are optional):
      - List Item 1:
        - source_ip_network: 1.1.1.0/24
        - destination_ip_network: 2.2.2.0/24

    - protocol: “1”
    - action: “DROP”
- List Item 2:
    - source_ip_network: “1.1.1.0/24”
    - destination_ip_network: “2.2.2.0/24”
    - action: “ACCEPT”
- List Item 3:
    - source_ip_network: “3.3.3.0/24”
    - destination_ip_network: “4.4.4.0/24”
    - source_port: “4500”
    - destination_port: “4800”
    - action: “BYPASS”

***NOTES**:*
The order of traffic selectors is important. They are provisioned in the order provided by an SA into the firewall. If more than one rule matches a packet, the first rule in the list will be used by the device.
In addition to IPsec service rules, the device creates FW rules for Syslog service and for configured CRL Distribution Points (CDPs). The default FW rules will block any ingress or egress flows unless they are part of the configured and enabled Syslog, CDP, RMI, IPsec services, or explicitly configured in the FW by an SA.

The configuration above would provision the device’s FW to allow ingress plaintext (PT) traffic from the device’s PT interface with the source IP address matching the configured “source_ip_network”. It will also allow egress of PT traffic to the PT interface with the destination IP address matching the configured “source_ip_network”. The “source_ip_network” parameter always refers to the device PT interface. The configuration above will also allow ingress of plaintext traffic from the PT interface that has the destination IP address matching the configured “destination_ip_network”. The configuration will also allow egress of plaintext traffic to the PT interface that has the destination IP address matching the configured “destination_ip_network”. The “destination_ip_network” parameter always refers to the IPsec Gateway (GW) peer’s PT interface.

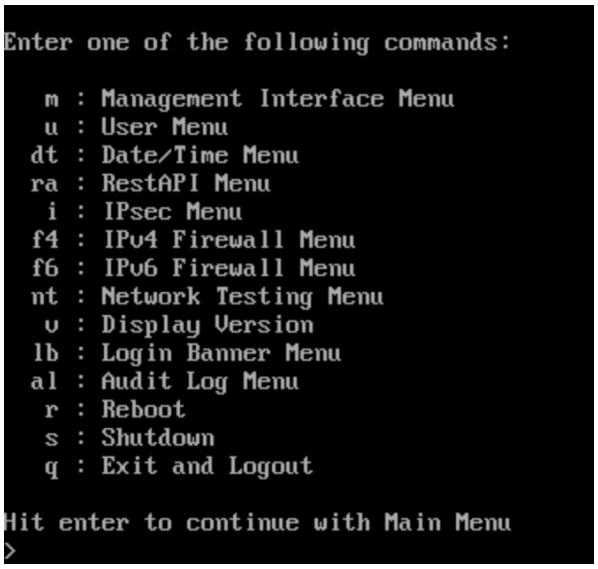
For situations where less typical and more elaborate packet filtering is required, explicit FW configuration described in section 7.2.7 is supported by the device.

7.2.7 Configuring FW Ruleset

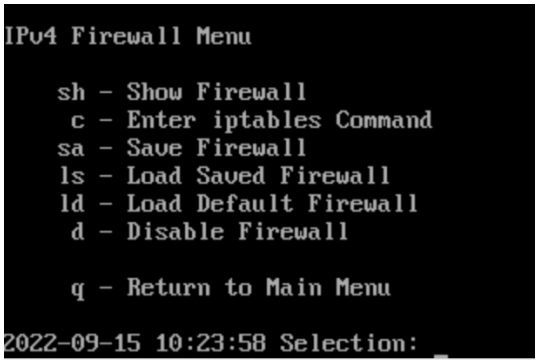
Firewall configuration is accessible from the CLI. Where IPsec Traffic Selector configuration does not fully produce the desired traffic flow policy enforcement, explicit Linux FW rules can augment the Traffic Selector configuration. The steps for FW rule configuration are as follows:

1. If not logged in, login to the device as described in section 4.1.

2. The main CLI menu will display. From the main CLI menu, type “f4” to open the IPv4 Firewall Menu or “f6” to open the IPv6 Firewall Menu.



3. FW menu is displayed (IPv4 and IPv6 FW sub-menus are identical. The IPV4 sub-menu is shows for as an example):



4. In the Firewall Menu, the following commands are available (IPv4 and IPv6 FW sub-menus are identical):
   - “sh” – Show Firewall
     - Display provisioned FW rules
   - “c” – Enter iptables command[21]
     - Allows a Security Administrator to enter FW commands using Linux iptables command syntax[22]
   - “sa” – Save Firewall

[21] Enter an iptables or ip6tables command after first entering “c” and device displaying a prompt.
[22] There are many resources available that describe iptables command syntax. This is one example: https://man7.org/linux/man-pages/man8/iptables.8.html; A reference is also provided in 9.1Appendix B.

- Persists the FW rules entered since last save request.

- "ls" – Load Saved Firewall
  - Loads previously saved FW rules
    - Note that saved FW rules are loaded automatically upon boot. This command is helpful when a Security Administrator changed the rules, has not saved them and wants to revert back to the saved configuration.
- "ld" – Load Default Firewall
  - Loades default FW rules.
- "d" – Disable Firewall
  - Disables the FW. This command must be used with extreme caution as it removes all FW protections. It is intended as a troubleshooting tool used for integration and not used in a fielded system.

***NOTES**: IPv6 Traffic Processing rules supported are Block and Bypass only. They must be configured directly in IPv6 Firewall settings. The device does not support IPsec encryption of IPv6 packets.*

IPv6 Firewall is configured by default to block access to the RMI using IPv6. IPv6 access to the RMI is not a NIAP-approved or secure configuration. IPv6 access to the RMI must be blocked to prevent a risk of CVE-2023-1206 Denial of Service (DoS) exploit.

The device automatically creates Firewall (FW) rules for its services, including Syslog client and RMI. These rules should not be altered by a Security Administrator.

When IPsec service is started, the Firewall (FW) rules required for basic IPsec operation are added to the iptables FW. When the service is stopped, such auto-generated FW rules are removed from the iptables FW. It must be recognized that when an SA adds rules to the FW manually and then performs IPsec service start/stop operations, the manually entered rules will remain, but the IPsec service auto-generated rules will be removed from their current FW rule ordering and added to the end of the FW rule chain. The SA can use manual FW management described above to change the order using iptables "-D, --delete" and "-I, --insert" commands.

The Traffic Processing rules include the implicit deny by default rule. If no other rules match the traffic, the traffic is blocked. Logging for the implicit deny rule is disabled by default to prevent large number of audit records generated. To enable logging for the implicit rule, add the log rule as the last rule in the iptables policy:

- *iptables -A INPUT -j LOG –log-prefix "IPT" -s X -d Y*

7.2.7.1 Firewall Rule Examples

Several common examples of IPv4 firewall rules are shown below to illustrate configuring IPsec packet processing rules using the FW function. Each rule is configured while in the IPv4 or IPv6 sub-menu described above, using the "c" – Enter iptables command option.

A FW rule is composed of a few key components. First, a Security Administrator (SA) needs to determine if the rule should be applied to one of the following network flows (using the -A switch):

- -A INPUT
  - Applies the rule to the INPUT firewall
  - The INPUT FW controls traffic flow into a network interface, i.e., the ingress flow.
- -A OUTPUT
  - Applies the rule to the OUTPUT firewall

- The OUTPUT FW controls traffic flow out of a network interface, i.e., the egress flow.

- -A FORWARD
  - Applies the rule to the FORWARD firewall
  - The FORWARD FW controls traffic flow from a network interface to an output network interface. This traffic is not protected by the vND IPsec function.

Second, an SA needs to determine what action should be applied to the rule (using the -j switch)

- -j ACCEPT
  - Allow the packet to pass through the FW function
  - NOTE: When FW is configured to accept a packet, the IPsec Traffic Policy implemented by the Security Policy Database (SPD) will determine what happens with the packet.
    - If the packet matches the SPD Policy for a Security Association (SA), the packet will be encrypted by the IPsec function and sent to the configured peer IPsec GW.
    - For simple PROTECT traffic policies, it is more convenient to use RMI Traffic Selector configuration described in section 7.2.6.
- -j DROP
  - Drop the packet
- -j LOG
  - Create an audit event every time the rule is matched.
  - Note that this rule needs to include the log prefix of “IPT” to have the rule generate an audit event.
    - Example: -j LOG -log-prefix “IPT”

Optionally, a rule can be applied to a specific protocol (using -p)

- -p <protocol> (icmp for example)

Optionally, you can apply the rule to a specific source IP address / IP network (using -s)

- -s <address[/mask]>

Optionally, you can apply the rule to a specific destination IP address / IP network (using -d)

- -d <address[/mask]>

Optionally, for protocols that support ports, you can apply the rule to a specific source port (using --sport)

- --sport 80

Optionally, for protocols that support ports, you can apply the rule to a specific destination port (using --dport)

- --dport 80

Optionally, you can apply the rule to a specific interface (using -i)

- -i eth1
- Note:
  - Management Interface – eth0
  - PT Interface – eth1
  - CT Interface – eth2

Examples:

- Add log rule for incoming IPv4 ICMP packets from source IP address X to destination IP address Y
  - To create packet log rules in iptables, they MUST include a prefix of “IPT”. E.g., log all packets that are TCP:
    - iptables -A INPUT -p icmp -j LOG –log-prefix “IPT” -s X -d Y
    - These logs rules have to be higher priority than other firewall rules matching the target packets.
  - This results in an example audit event:
    - 2022-08-30T16:27:34+00:00 VSVPN kernel: [6272.607648] "(AUDIT_LOG_EVENT) Event_Type: iptables LOG rule match, Event_Outcome: Success, Event_Functional_Component: match "IN=eth0 OUT= MAC=00:0c:29:a7:6f:68:3e:22:fb:68:f0:64:08:00 SRC=X DST=Y LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=47005 PROTO=ICMP TYPE=8 CODE=0 ID=19711 SEQ=1
- Block IPv4 packets on the PT interface for protocol tcp and destination IP network 1.1.1.0/24 using port 4500
  - Iptables -A INPUT -p tcp --sport 4500 --dport 4500 -d 1.1.1.0/24 -i eth1 -j DROP
- Block IPv6 packets on the PT interface for protocol udp and destination IP address range 2.2.2.0/24
  - Iptables6 -A INPUT -p udp -d 2.2.2.0/24 -i eth1 -j DROP
- Bypass all IPv6 icmp packets from PT to CT
  - Iptables6 -A FORWARD -i eth1 -o eth2 -p ipv6-icmp -j ACCEPT
  - Iptables6 -A FORWARD -i eth2 -o eth1 -p ipv6-icmp -m state --state ESTABLISHED,RELATED -j ACCEPT
  - Iptables6 -t nat -A POSTROUTING -o eth2 -j MASQUERADE

7.3 PlainText (PT) Interface Settings

The PT interface is dedicated to ingesting plaintext user traffic from the local network and delivering plaintext user traffic to a local network destination that is received from a peer VPN GW and decrypted by the device

PT interface network configuration can be viewed and modified using the RMI as follows:

- View PT Interface network settings:
  - **`GET /network/plaintext`**
- Set PT Interface network settings:
  - **`POST /network/plaintext`**
    - Parameters: restart_network:boolean, whether to restart the network interface and apply changes immediately.
    - Body:

- ipv4_network (IPv4 address for this interface in CIDR format):string
- ipv4_gateway (IPv4 address of the default gateway for this interface):string
- ipv6_network (IPv6 Address for this interface in CIDR format):string
- ipv6_gateway (IPv6 address of the default gateway for this interface):string

7.4 CipherText (CT) Interface Settings

The CT interface supports up to four physical or virtual network interfaces that can be configured in the vND. The device always uses a single CT interface at the time but would automatically failover to the next CT interface in the order specified by the Security Administrator (SA) when a failure detected with the current CT interface. Note that NIAP evaluation only covered using a single CT interface and did not include the failover capability.

Since the device is a VM, each CT interface whether it maps to a physical Network Interface Card (NIC) on the hardware platform or is a virtual only NIC (vNIC), must be configured in the Virtual Machine (VM) as a virtual network adapter using Hyper-V. As noted earlier, CT interfaces must appear as Eth2 through Eth7 inclusive in the VM.

Each CT interface's network configuration can be viewed and modified using the RMI as follows:

- View CT Interface network settings:
  - **`GET /network/ciphertext/{interface_id}`**
    - Where **`interface_id`** is from 1 to 6 and indicates a CT interface number in the order it was configured for the VM in Hyper-V
- Set CT Interface network settings:
  - **`POST /network/ciphertext/{interface_id}`**
    - Where **`interface_id`** is from 1 to 6 and indicates a CT interface number in the order it was configured for the VM in Hyper-V
    - Parameters: restart_network (whether to restart the network interface and apply changes immediately):boolean
    - Body:
      - ipv4_network (IPv4 address for this interface in CIDR format):string
      - ipv4_gateway (IPv4 address of the default gateway for this interface):string
      - ipv6_network (IPv6 Address for this interface in CIDR format):string
      - ipv6_gateway (IPv6 address of the default gateway for this interface):string

***NOTES**:*
An example interface ID: if the VM has eth2 and eth3 interfaces configured at the Hyper-V level for CT, the eth2 is interface ID 1 and eth3 is interface ID 2.
An example of IPv4 device CT address in CIDR format: 8.37.45.14/24

CT interface priority is used to allow the device to failover from a higher precedence CT interface to a lower should the higher precedence CT interface experience a failure. The failure is detected by the

device when the underlying platform reports that the interface lost its physical or datalink connection. The CT interface priority is configured using the RMI as follows:

- View CT network interface priority:
  - **GET /network/link_preference**
- Set CT network interface priority:
  - **POST /network/link_preference**
    - Parameters:
      - link_preference_type ({automatic|manual}):string
      - Request body (a list of CT interface IDs, separated by comas)
        - Example: [1,2]

***NOTE**:*
An example interface ID: if the VM has eth2 and eth3 interfaces configured at the Hyper-V level for CT, the eth2 is interface ID 1 and eth3 is interface ID 2.

7.5 Domain Name System (DNS) Configuration

The device does not query an external DNS service for Domain Name (DN) resolution. Instead, a Security Administrator (SA) can configure domain name to IP address mapping in the device. Such mapping would be used when Trusted Channel (TC) and Trusted Path (TP) interfaces connections are made to external systems. It is also used to retrieve Certificate Revocation Lists (CRLs) that are usually downloaded from a domain name specified CRL Distribution Point (CDP), defined in X.509 certificates.

When an SA configures domain name to IP address mapping in the device, as described below, the device also provisions firewall (FW) rules to allow a connection to the IP address specified by the SA.

DN resolution mapping is configured through the RMI as follows:

- View DNS mapping configuration:
  - **GET /network/dns_entries**
- Set DNS mapping:
  - **POST /network/dns_entries**
    - Parameters:
      - Request body (a list of DN to IP mapping, separated by commas)
        - Example:

[{"domain_name": "mail.server.com", "ip_address": "198.51.100.42" },

{"domain_name": "cdp.server1.net", "ip_address": "198.51.200.42" },

{"domain_name": "syslog.server2.us", "ip_address": "198.51.300.42"}]

***NOTE**: Setting DN mapping removes all old mappings previously configured. To retain prior mappings, execute the GET operation, copy the results, add/modify the mappings from the GET operation, then use the updated list in the POST operation.*

8 ADDITIONAL DEVICE CONFIGURATION ITEMS

Table 8-1: Device Configuration Items (Cis) Summary

#	CI	Description	CLI/ RMI	Comments	Described
1	Login Banner Text	Set the login banner text for CLI and RMI.	CLI	The banner text is configured in the CLI	8.2 - Configuring Logon Banner
2	CLI Inactivity Timeout	Set, review CLI inactivity timeout	CLI	Range is from 3 to 120 minutes	8.5 – Configuring SA Login Session Timeout
3	RMI Inactivity Timeout	Set, review RMI inactivity timeout	CLI	Range is from 3 to 120 minutes	8.5 – Configuring SA Login Session Timeout
4	Software Update	Initiate device secure software update process.	RMI	The device validates a digital signature over the software image prior to installation.	8.8 – Software Update
5	SA Login Authenticatio n Failure Actions	Definition of the number of the failed successive RMI logins before login timeout is enabled. Also, the value of the timeout.	RMI	Range of the failed login attempts is from 3 to 10. Time out value range is from 30 to 1200 seconds.	8.6 User Failed Login Handling
6	Key Management	Includes: -Generating PKI key-pair, downloading a CSR. -Loading a device PKI certificate. -Zeroizing PKI keys and certificate -Loading a PKI Trust Anchor -Deleting a PKI Trust Anchor -Loading Intermediate CA certificate -Deleting Intermediate CA certificate	RMI	Note that separate KMAT stores are used for IPsec, HTTPS and Syslog.	6.1 – Device KMAT Management
7	Cryptographi c Algorithm Settings	Includes: -ECC NIST Curve selection	RMI	Selected for IPsec TC only, upon device PKI certificate	6.1 – Device KMAT Management

#	CI	Description	CLI/ RMI	Comments	Described
				enrollment process.	
8	IPsec lifetime	Configuration of IKEv2 SA Lifetime and ESP lifetime	CLI	IKEv2 SA lifetime range is from 4hrs to 120hrs (5 days). ESP (AKA child) SA range is from 2hrs to 48hrs (2 days).	7.2.2 - IPsec Security Association Lifetime Configuration
9	Start/Stop VPN Service	Starting or stopping the IPsec VPN service	RMI		7.2.1 - IPsec Service Start/Stop
10	Security Audit: Syslog	Delivery of device security audit records to a Syslog server, including: -Syslog server IP Address and Domain Name -Syslog server port (or default) -TLS enable/disable	RMI	Default TCP port is 6514 (with TLS).	5.1.1 – Syslog Service Management
11	Device Time	Set device time	CLI		8.9 – Time Management
12	IPsec Peer Reference Identifier	Review and configure peer reference identifier for IPsec authentication.	RMI	Only a peer with a matching identifier in the certificate is allowed to establish IPsec connection.	7.2.3 - IPsec Peer Reference Identifier
13	Syslog Server Reference Identifier	Review and configure a server reference identifier for Syslog authentication.	RMI	Only servers with matching identifiers in the certificate are allowed to establish a Trusted Channel.	5.1.1 – Syslog Service Management
14	IPsec Packet Filtering Rules	Definition of IPsec packet filtering rules to define matching network flow to allow/drop operation and a network interface. Also ordering packet filtering rules by	CLI, RMI		7.2.4 - IPsec Device and Peer Port

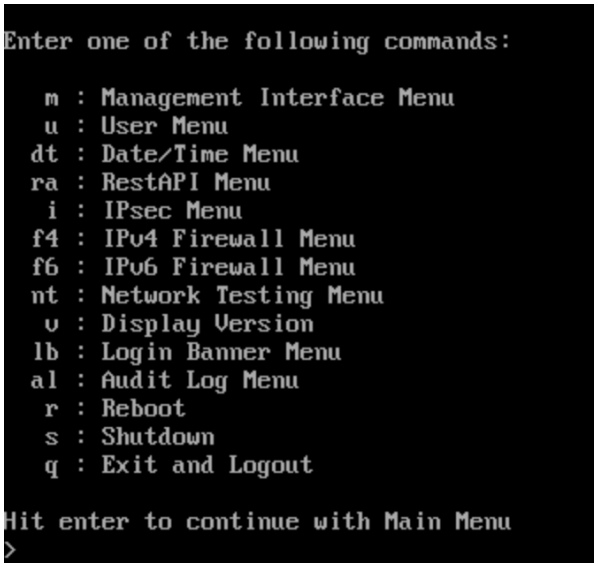
#	CI	Description	CLI/ RMI	Comments	Described
		priority			
15	IPsec Remote Session Timeout	Remote session (AKA lifetime) timeout to terminate an IPsec session upon reaching the configured timeout value.	CLI	IKE lifetime range is 4-120 hours. The default is 24 hours. ESP SA lifetime range is 2-48 hours. The default is 8 hours.	7.2.2 - IPsec Security Association Lifetime Configuration

8.1 Network Test Functions

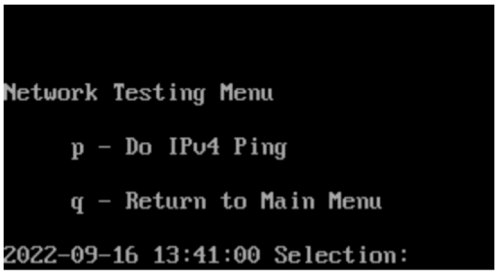
The device provides rudimentary network testing functions that can assist a Security Administrator (SA) in troubleshooting connectivity errors. This capability is not included in NIAP evaluation.

To perform network testing:

1. While an SA is logged into the device CLI, from the main menu:

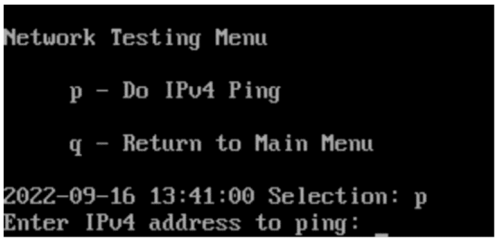


2. Type “nt” to open Network Testing Menu.
3. The device displays the menu:



4. Select “p” to perform an IPv4 ping

5. At the device prompt shown below, enter a valid IPv4 address and hit enter.

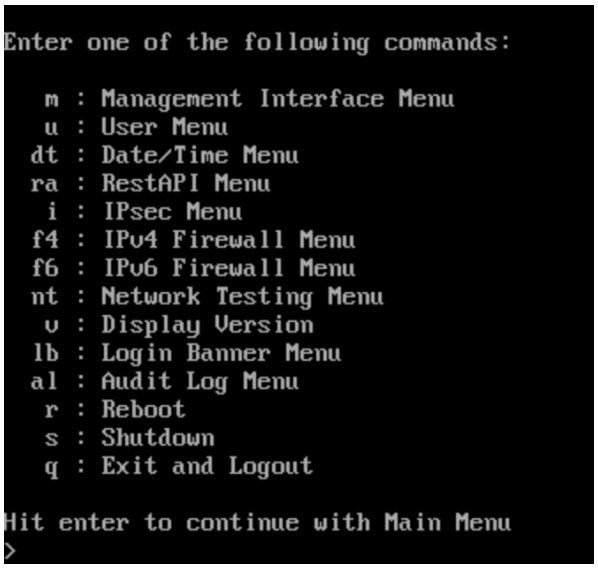


6. The device performs the requested ping.

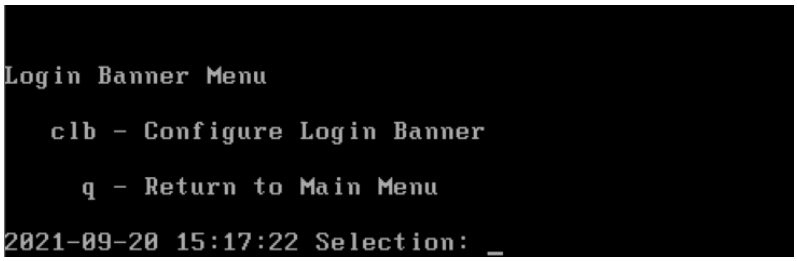
8.2 Configuring Logon Banner

The logon banner is displayed upon CLI and RMI operator access. The text is configurable using the CLI as follows:

1. If not logged in, login to the device as described in section 4.1.
2. The main CLI menu will display. From the main CLI menu, type “lb” to open the Login Banner Menu.



3. From the Login Banner Menu, type “clb” to Configure Login Banner.

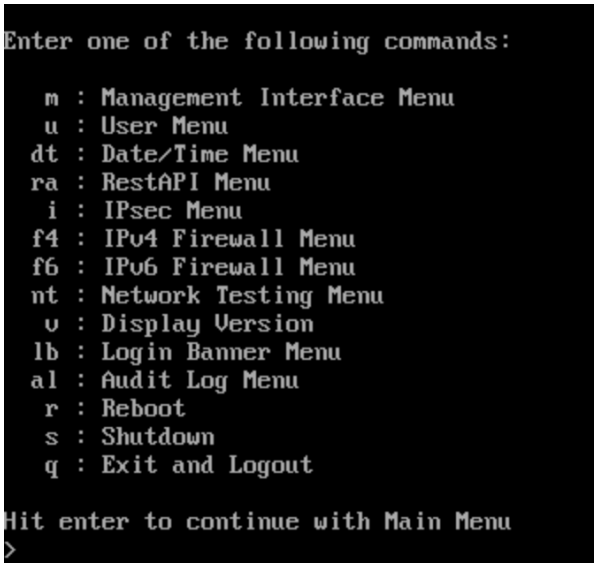


4. Enter new login banner text.
5. Enter "!" character to save the changes.[23]

8.3 Configuring Minimum Password Length

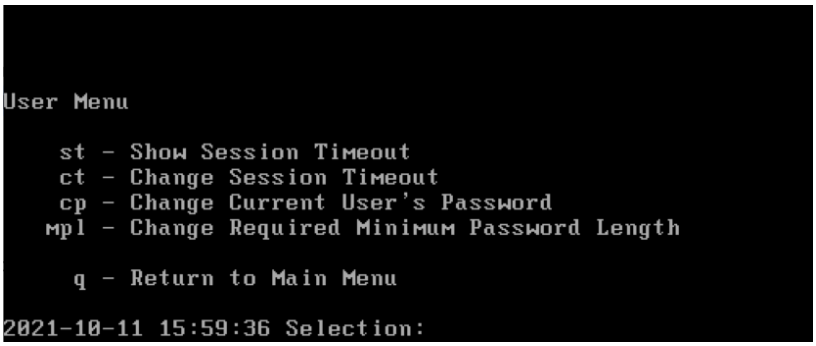
The password minimum length enforcement is configurable from 6 to 20 characters through the CLI. The following steps describe that process:

1. If not logged in, login to the device as described in section 4.1.
2. The main CLI menu will display. From the main CLI menu, type "u" to open the User Menu.



3. In the User Menu, select "mpl" for Change Required Minimum Password Length.

[23] The "!" character terminates login banner entry. It cannot be used in the text of the banner.

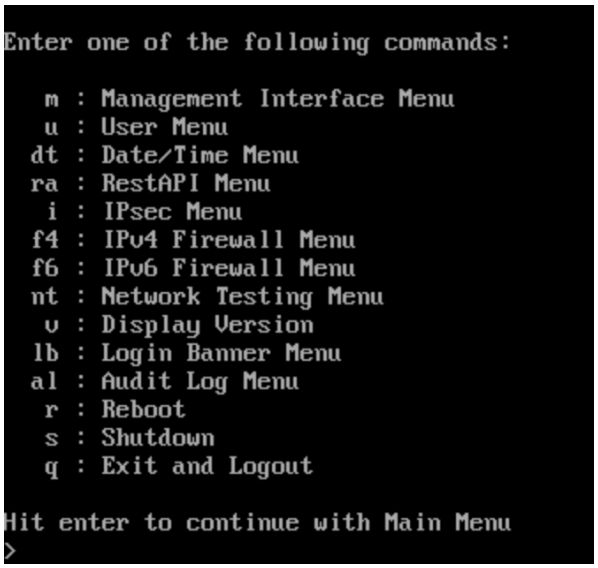


4. At the prompt, enter a password length within the range of 6 to 20 characters and hit Enter.

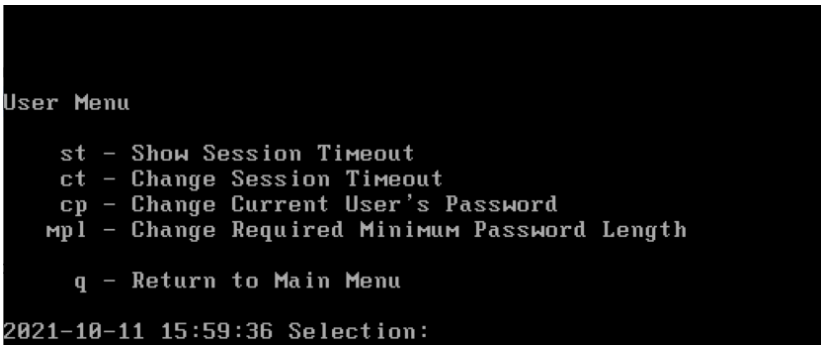
8.4 Configuring SA Password

The password is configurable from the CLI and the new password must meet the password composition and minimum password length requirements noted in section 4.1.3 - Security Administrator (SA) Login Session. The following steps describe SA password change:

1. If not logged in, login to the device as described in section 4.1.
2. The main CLI menu will display. From the main CLI menu, type “u” to open the User Menu.



3. In the User Menu, type “cp” to Change Current User’s Password.



4. At the prompts, enter the new password twice and hit Enter. If the passwords match and meet password requirements (e.g. length, characters), the password will be updated for the CLI and RMI SA access.

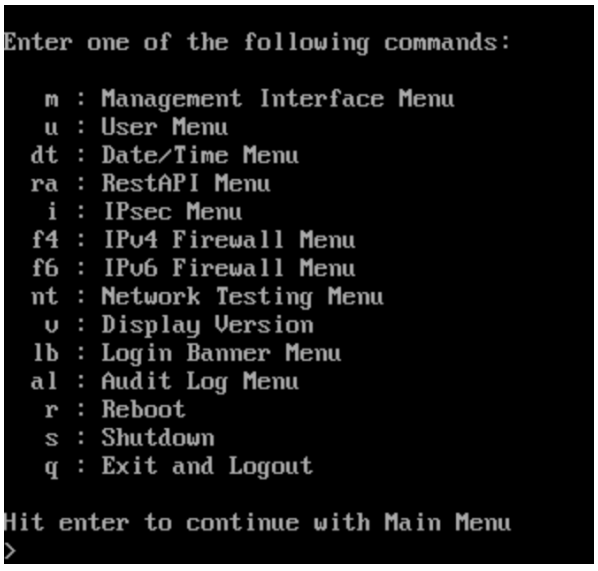
8.5 Configuring SA Login Session Timeout

The SA login session timeout feature is intended to mitigate session hijacking when a privileged login session is left unattended. After a login session is left idling for a predefined time period configured here, the session is terminated by the device.

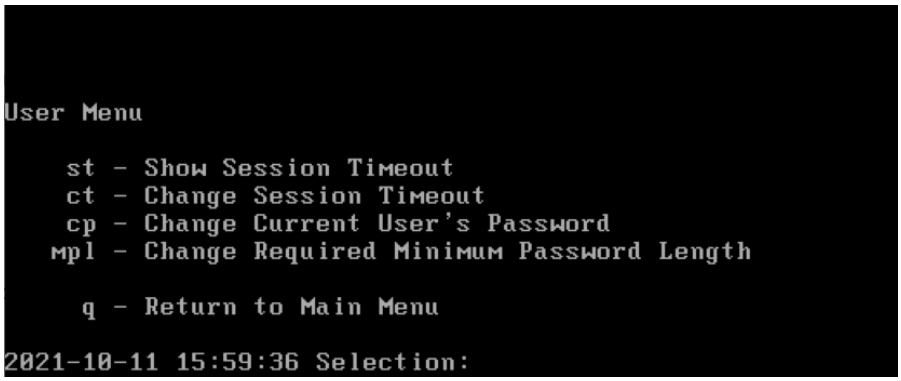
The CLI session timeout is configured through the CLI and the RMI session timeout is configured through the RMI.

The following steps describe SA session timeout setting for the CLI:

1. If not logged in, login to the device as described in section 4.1.
2. From the main CLI menu, type “u” to open the User Menu.



3. While in the User Menu, type “ct” to Change Session Timeout



4. At the prompt, enter a timeout value within the allowed range of 30 to 7,200 seconds and hit Enter.

To configure the RMI session timeout, the following REST API commands are used:

1. Read the value:
   a. **GET /system/inactivity_timeout**
2. Set the value:
   a. **POST /system/inactivity_timeout**
      i. Parameters: timeout_sec (30-7200):integer

8.6 User Failed Login Handling Configuration

The device supports locking the user account after a configurable number of failed login attempts to prevent brute force password attacks against the Security Administrator (SA) account. The account is locked at the Remote Management Interface (RMI) only, while the account is always accessible through the Local Command Interface (CLI). The account remains locked for a configurable time period.

To configure the number of successive failed RMI login attempts, the following RMI commands are used:

1. Reading the value:
   a. **GET /auth/login_attempts**
2. Setting the value:
   a. **POST /auth/login_attempts**
      i. Parameters: login_attempts (from 3 to 10) : integer

To configure the delay timeout value, the following RMI commands are used:

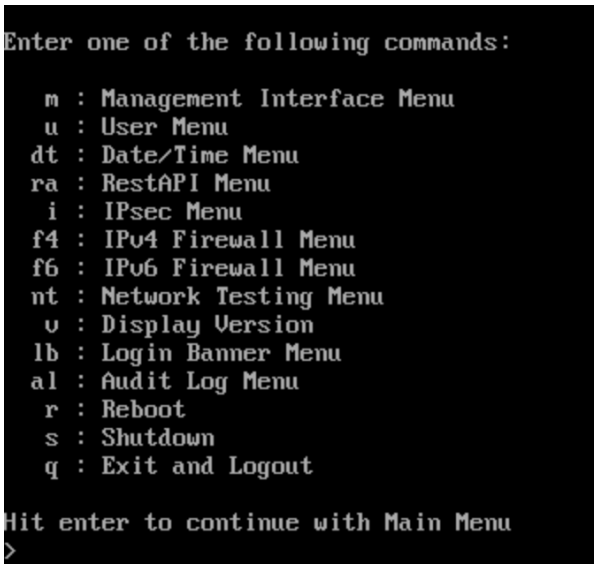
1. Reading the value
   a. **GET /auth/login_timeout**
2. Setting the value:
   a. **POST /auth/login_timeout**
      i. Parameters login_timeout (from 60 to 3600) (in seconds) : integer

8.7 Deleting Internally Stored Audit Records

An SA can request deletion of audit records from the internal audit store. After the deletion, audit records that have not been forwarded by the device to an external Syslog service earlier will be lost.

The following steps describe audit log deletion:

1. If not logged in, login to the device as described in section 4.1.
2. From the main CLI menu, type “al” to open the Audit Log Menu



3. While in the Audit Log Menu, type “d” to Delete Audit Logs.

4. The device performs the requested operation.

***NOTE**: If the audit logs have not been delivered to a remote Syslog service, they will be lost.*

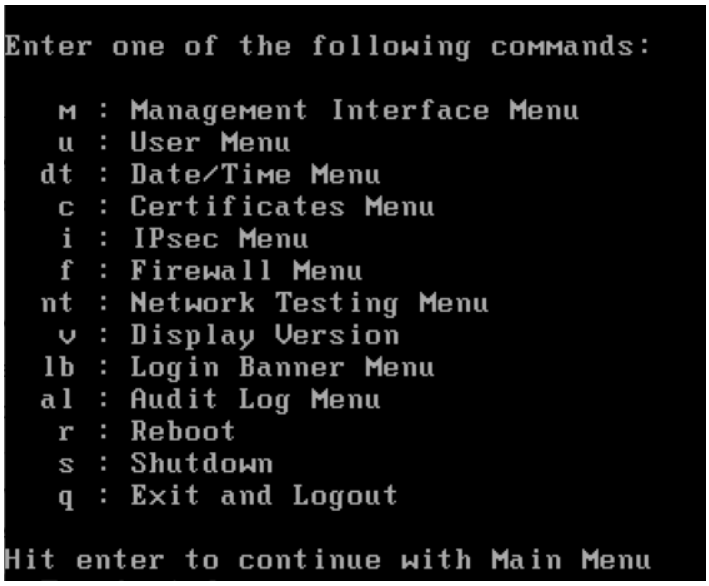
8.8 Software Update

The device provides means to query the running software version and to perform a secure software update. The secure update process requires that the new software image is digitally signed as provided by Viasat. Upon the upgrade, the digital signature is verified and if it is valid, the upgrade is performed, and an audit record is generated indicating success. Otherwise, the upgrade is aborted and an audit record is generated indicating failure.

To obtain the device update package, login to your Viasat customer portal account as described in section 2.4. Follow the instructions in the portal to access your purchased Viasat products and download the vND update package for “Viasat Secure VPN”.

To query the running software version, using the CLI:

1. If not logged in, login to the device as described in section 4.1.
2. From the CLI main menu, select “v” for “Display Version” menu item.



3. The device displays running software version.
4. Press Enter to return to the main menu.

To perform the secure upgrade, using the RMI:

Uploading a new software image:

1. PUT /system/upgrade
   - Parameters: keep_settings (Boolean: {True|False}), indicates whether to keep device configuration, audit logs and KMAT after the upgrade.
   - Body: bin_file:binary, the software image file to upload and use for upgrade
2. Upon receiving this request, the device validates the image, including performing image digital signature validation.
   a. Digital signature validation includes validation of the certificate trust path and status validation as described in section 6.3. If certificate status validation fails for any reason, the image is deemed invalid, and the upgrade is aborted.
   b. If the digital signature checks complete successfully, the image is in correct format and the version being upgraded to is greater than the current version, the device loads and persists the new image.
3. After the new version is loaded, the device restarts automatically to start running the new version.

NOTES:

Upon request to perform a software update, all vND services will continue to execute normally until the update is loaded into the device and successfully validated. At that point, the VND will shutdown all services and will restart with the new image. If retain settings option has been selected, the device will attempt to restart all services that were operating prior to the update.

After the update is requested, the vND will not further prompt the Security Administrator (SA). If the update passes all the checks, the device It will restart and activate the new software automatically.

CRL download or CRL validation failure during the secure software update will result in

upgrade abort. To recover, obtain an update image with valid digital signature and ensure that CRLs are available for download from the specified CRL Distribution Points (CDPs).

8.9 Time Management

The device uses its internal clock to maintain system time. As a virtual Network Device (vND), it can rely on the underlying hardware platform and the Hyper-V hypervisor or Security Administrator (SA) to maintain time between VM pause and resume, shutdown and restart events.

The device supports using the underlying Hyper-V VM time synchronization feature and the feature being disabled with the cognizant SA performing time management activities manually. Enabling and disabling the Hyper-V VM time synchronization feature is described in section 2.3.

When the Hyper-V VM time synchronization is not used, the device time can be set by an authorized SA using the CLI or the RMI as described below. It must be noted that the device will maintain the time while the device is running. Upon shutting down or pausing the device's VM execution, the time will be incorrect and must be configured again by the SA.

When Hyper-V VM time synchronization feature is enabled, the Hyper-V will synchronize the device's clock with the Windows 10 host OS time. However, the time synchronization only occurs when the device's VM is paused and resumed using the Hyper-V. Depending on the time difference between the Windows OS and the device's VM, reboot may or may not synchronize the VM to the Windows OS time. In all cases, the SA may use the CLI or the RMI as described below to verify that the device's time is set correctly.

The procedure for manually setting the device's time is as follows:

1. If not logged in, login to the device as described in section 4.1.
2. From the CLI main menu, select "Date/Time Menu"

```
Enter one of the following commands:

   m : Management Interface Menu
   u : User Menu
  dt : Date/Time Menu
   c : Certificates Menu
   i : IPsec Menu
   f : Firewall Menu
  nt : Network Testing Menu
   v : Display Version
  lb : Login Banner Menu
  al : Audit Log Menu
   r : Reboot
   s : Shutdown
   q : Exit and Logout

Hit enter to continue with Main Menu
```

3. If the SA needs to change the time, from the Date/Time Menu, select “Change Date/Time”

```
Date/Time Menu

     cdt - Change Date/Time
           Recognized TIME formats:
           hh:mm[:ss]
           [YYYY.]MM.DD-hh:mm[:ss]
           YYYY-MM-DD hh:mm[:ss]
           [[[[[YY]YY]MM]DD]hh]mm[.ss]

     q - Return to Main Menu

2021-10-08 09:37:16 Selection:
```

4. Enter new date and time in the format indicated, followed by enter.

***NOTES**: While several formats are supported as indicated by the CLI, the recommended format for entering system date and time is YEAR.MONTH.DAY-HOUR:MINUTE:SECOND. The time must be in the UTC.*

vND time can also be managed through the RMI as follows:

1. Read system time:
    a. **GET /system/date**, get system time
2. Set system time:
    a. **POST /system/date**, sets system time
        i. Parameters: date-time:string, in the following format YYYY-MONTH-DAY HOUR:MIN:SEC.

8.10 System Failure Recovery

Upon system failure (crush, power loss, etc.), the device can be restarted and should resume normal operation. In situations where the failure happed during SA configuring the device, the last configuration step before the failure may need to be repeated.

If the device fails to start multiple times after a failure, it is possible that the virtual drive and the device software may have been corrupted by the failure. It is recommended that the Viasat Secure VPN software installation be repeated as described in section 2.4.

Should the problem persist, contact Viasat Customer Case Center.

9 SELF-TEST

The device performs security self-tests upon start up. The following self-tests are performed:

- Cryptographic algorithm test to verify correct operation for each supported algorithm.
  - Upon virtual Network Device (vND) startup
- Verification of integrity of software
  - Upon vND startup
- Random Number Generator (RNG) entropy source noise tests.
  - Upon physical machine startup

The cryptographic algorithm testing includes execution of each supported by the device algorithm and given a known input, verification of the corresponding known answer. This is referred to as a Known Answer Test (KAT).

The verification of the integrity of the software ensures that the device software image matches the software image that was cryptographically authenticated and installed in the Virtual Machine (VM). These tests involve using image digest checks to ensure the software image has not been altered, corrupt or otherwise modified.

The certified device configuration uses Intel Secure Key technology. This technology provides a NIST SP 800-90 A, NIST SP 800-90 B and NIST SP 800-90 C compliant seed and RNG implementation. The entropy hardware in the 11th Gen Intel Core i5-1140 @ 2.6Ghz processor performs the required entropy noise tests. The Intel® Secure Key technology in the processor performs Built-In Self Tests (BISTs) to verify the correct operation of the Entropy Source (ES) prior to making the Digital Random Number Generator (DRNG) available to hosted software. The device verifies that the BIST was successful prior to using processor provided DRNG data.

Upon a failure of any of the self-tests above, the device provides a notification to a Security Administrator using the CLI and then shuts down.

9.1 Self-Test Failure Recovery

As noted earlier, any self-test failure would result in vND shutting down (fail-secure). All self-tests must successfully complete prior to the vND transitioning to its operational state.

Should any of the self-tests fail the following recovery is recommended:

- Cryptographic algorithm test to verify correct operation for each supported algorithm.
  - Restart the vND up to two times and observe if that resolves the problem.
  - If the problem persists, reinstall the vND as described in section 2.1.
  - If the problem persists, contact Viasat Customer Care Center as described in section 3.
- Verification of integrity of software
  - Restart the vND up to two times and observe if that resolves the problem.
  - If the problem persists, reinstall the vND as described in section 2.1.
  - If the problem persists, contact Viasat Customer Care Center as described in section 3.
- Random Number Generator (RNG) entropy source noise tests.
  - Power cycle/reboot the physical platform up to two times and observe if that resolves the problem.
  - If the problem persists, contact the hardware manufacturer for resolution.

Appendix A RMI REST API REFERENCE

Full REST API Reference is available from Viasat Customer Support Portal (noted earlier in this document). The reference provides a human readable form of the RMI REST API interfaces (AKA endpoints). In addition, the Appendix A contains JSON listing for the REST API interface that can be loaded into REST API tools to access the vND RMI. See file: 1388812_AppA_Viasat Secure VPN, version 1.1.7, REST API Reference.docx

Appendix B LINUX IPTABLES REFERENCE[24]

B.1 NAME

iptables/ip6tables — administration tool for IPv4/IPv6 packet filtering and NAT

B.2 SYNOPSIS

iptables [**-t** *table*] {**-A**|**-C**|**-D**} *chain rule-specification*
ip6tables [**-t** *table*] {**-A**|**-C**|**-D**} *chain rule-specification*
iptables [**-t** *table*] **-I** *chain* [*rulenum*] *rule-specification*
iptables [**-t** *table*] **-R** *chain rulenum rule-specification*
iptables [**-t** *table*] **-D** *chain rulenum*
iptables [**-t** *table*] **-S** [*chain* [*rulenum*]]
iptables [**-t** *table*] {**-F**|**-L**|**-Z**} [*chain* [*rulenum*]] [*options...*]
iptables [**-t** *table*] **-N** *chain*
iptables [**-t** *table*] **-X** [*chain*]
iptables [**-t** *table*] **-P** *chain target*
iptables [**-t** *table*] **-E** *old-chain-name new-chain-name*
rule-specification = [*matches...*] [*target*]
match = **-m** *matchname* [*per-match-options*]
target = **-j** *targetname* [*per-target-options*]

B.3 DESCRIPTION

Iptables and **ip6tables** are used to set up, maintain, and inspect the tables of IPv4 and IPv6 packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains.

Each chain is a list of rules which can match a set of packets. Each rule specifies what to do with a packet that matches. This is called a `target', which may be a jump to a user-defined chain in the same table.

B.4 TARGETS

A firewall rule specifies criteria for a packet and a target. If the packet does not match, the next rule in the chain is examined; if it does match, then the next rule is specified by the value of the target, which can be the name of a user-defined chain, one of the targets described in iptables-extensions(8), or one of the special values **ACCEPT**, **DROP** or **RETURN**.

ACCEPT means to let the packet through. **DROP** means to drop the packet on the floor. **RETURN** means stop traversing this chain and resume at the next rule in the previous (calling) chain. If the end of a built-in chain is reached or a rule in a built-in chain with target **RETURN** is matched, the target specified by the chain policy determines the fate of the packet.

[24] Obtained from https://man7.org/linux/man-pages/man8/iptables.8.html.

B.5 TABLES

There are currently five independent tables (which tables are present at any time depends on the kernel configuration options and which modules are present).

-t, --table
This option specifies the packet matching table which the command should operate on. If the kernel is configured with automatic module loading, an attempt will be made to load the appropriate module for that table if it is not already there.

The tables are as follows:

filter: This is the default table (if no -t option is passed). It contains the built-in chains **INPUT** (for packets destined to local sockets), **FORWARD** (for packets being routed through the box), and **OUTPUT** (for locally-generated packets).

nat: This table is consulted when a packet that creates a new connection is encountered. It consists of four built-ins: **PREROUTING** (for altering packets as soon as they come in), **INPUT** (for altering packets destined for local sockets), **OUTPUT** (for altering locally-generated packets before routing), and **POSTROUTING** (for altering packets as they are about to go out). IPv6 NAT support is available since kernel 3.7.

mangle: This table is used for specialized packet alteration. Until kernel 2.4.17 it had two built-in chains: **PREROUTING** (for altering incoming packets before routing) and **OUTPUT** (for altering locally-generated packets before routing). Since kernel 2.4.18, three other built-in chains are also supported: **INPUT** (for packets coming into the box itself), **FORWARD** (for altering packets being routed through the box), and **POSTROUTING** (for altering packets as they are about to go out).

raw: This table is used mainly for configuring exemptions from connection tracking in combination with the NOTRACK target. It registers at the netfilter hooks with higher priority and is thus called before ip_conntrack, or any other IP tables. It provides the following built-in chains: **PREROUTING** (for packets arriving via any network interface) **OUTPUT** (for packets generated by local processes)

security: This table is used for Mandatory Access Control (MAC) networking rules, such as those enabled by the **SECMARK** and **CONNSECMARK** targets. Mandatory Access Control is implemented by Linux Security Modules such as SELinux. The security table is called after the filter table, allowing any Discretionary Access Control (DAC) rules in the filter table to take effect before MAC rules. This table provides the following built-in chains: **INPUT** (for packets coming into the box itself), **OUTPUT** (for altering locally-generated packets before routing), and **FORWARD** (for altering packets being routed through the box).

B.6 OPTIONS

The options that are recognized by **iptables** and **ip6tables** can be divided into several different groups.

COMMANDS
These options specify the desired action to perform. Only one of them can be specified on the command line unless otherwise stated below. For long versions of

the command and option names, you need to use only enough letters to ensure that **iptables** can differentiate it from all other options.

-A, --append *chain rule-specification*
Append one or more rules to the end of the selected chain. When the source and/or destination names resolve to more than one address, a rule will be added for each possible address combination.

-C, --check *chain rule-specification* Check whether a rule matching the specification does exist in the selected chain. This command uses the same logic as

-D to find a matching entry, but does not alter the existing iptables configuration and uses its exit code to indicate success or failure.

-D, --delete *chain rule-specification*
-D, --delete *chain rulenum* Delete one or more rules from the selected chain. There are two versions of this command: the rule can be specified as a number in the chain (starting at 1 for the first rule) or a rule to match.

-I, --insert *chain* [*rulenum*] *rule-specification* Insert one or more rules in the selected chain as the given rule number. So, if the rule number is 1, the rule or rules are inserted at the head of the chain. This is also the default if no rule number is specified.

-R, --replace *chain rulenum rule-specification* Replace a rule in the selected chain. If the source and/or destination names resolve to multiple addresses, the command will fail. Rules are numbered starting at 1.

-L, --list [*chain*] List all rules in the selected chain. If no chain is selected, all chains are listed. Like every other iptables command, it applies to the specified table (filter is the default), so NAT rules get listed by iptables -t nat -n -L Please note that it is often used with the **-n** option, in order to avoid long reverse DNS lookups. It is legal to specify the **-Z** (zero) option as well, in which case the chain(s) will be atomically listed and zeroed. The exact output is affected by the other arguments given. The exact rules are suppressed until you use iptables -L -v or iptables-save(8).

-S, --list-rules [*chain*] Print all rules in the selected chain. If no chain is selected, all chains are printed like iptables-save. Like every other iptables command, it applies to the specified table (filter is the default).

-F, --flush [*chain*] Flush the selected chain (all the chains in the table if none is given). This is equivalent to deleting all the rules one by one.

-Z, --zero [*chain* [*rulenum*]] Zero the packet and byte counters in all chains, or only the given chain, or only the given rule in a chain. It is legal to specify the **-L, --list** (list) option as well, to see the counters immediately before they are cleared. (See above.)

-N, --new-chain *chain* Create a new user-defined chain by the given name. There must be no target of that name already.

-X, --delete-chain [*chain*] Delete the optional user-defined chain specified. There must be no references to the chain. If there are, you must delete or replace the referring rules before the chain can be deleted. The chain must be

empty, i.e. not contain any rules. If no argument is given, it will attempt to delete every non-builtin chain in the table.

-P, **--policy** *chain target* Set the policy for the built-in (non-user-defined) chain to the given target. The policy target must be either **ACCEPT** or **DROP**.

-E, **--rename-chain** *old-chain new-chain* Rename the user specified chain to the user supplied name. This is cosmetic, and has no effect on the structure of the table.

-h Help. Give a (currently very brief) description of the command syntax.

B.7 PARAMETERS

The following parameters make up a rule specification (as used in the add, delete, insert, replace and append commands).

-4, **--ipv4** This option has no effect in iptables and iptables-restore. If a rule using the **-4** option is inserted with (and only with) ip6tables-restore, it will be silently ignored. Any other uses will throw an error. This option allows IPv4 and IPv6 rules in a single rule file for use with both iptables-restore and ip6tables-restore.

-6, **--ipv6** If a rule using the **-6** option is inserted with (and only with) iptables-restore, it will be silently ignored. Any other uses will throw an error. This option allows IPv4 and IPv6 rules in a single rule file for use with both iptables-restore and ip6tables-restore. This option has no effect in ip6tables and ip6tables-restore.

[**!**] **-p**, **--protocol** The protocol of the rule or of the packet to check. The specified protocol can be one of **tcp**, **udp**, **udplite**, **icmp**, **icmpv6**,**esp**, **ah**, **sctp**, **mh** or the special keyword "**all**", or it can be a numeric value, representing one of these protocols or a different one. A protocol name from /etc/protocols is also allowed. A "!" argument before the protocol inverts the test. The number zero is equivalent to **all**. "**all**" will match with all protocols and is taken as default when this option is omitted. Note that, in ip6tables, IPv6 extension headers except **esp** are not allowed. **esp** and **ipv6-nonext** can be used with Kernel version 2.6.11 or later. The number zero is equivalent to **all**, which means that you cannot test the protocol field for the value 0 directly. To match on a HBH header, even if it were the last, you cannot use **-p 0**, but always need **-m hbh**.

[**!**] **-s**, **--source** *address*[**/***mask*][**,...**] Source specification. *Address* can be either a network name, a hostname, a network IP address (with **/***mask*), or a plain IP address. Hostnames will be resolved once only, before the rule is submitted to the kernel. Please note that specifying any name to be resolved with a remote query such as DNS is a really bad idea. The *mask* can be either an ipv4 network mask (for iptables) or a plain number, specifying the number of 1's at the left side of the network mask. Thus, an iptables mask of *24* is equivalent to *255.255.255.0*. A "!" argument before the address specification inverts the sense of the address. The flag **--src** is an alias for this option. Multiple addresses can be specified, but this will **expand to multiple rules** (when adding with -A), or will cause multiple rules to be deleted (with -D).

[**!**] **-d**, **--destination** *address*[**/***mask*][**,...**] Destination specification. See the description of the **-s** (source) flag for a detailed description of the syntax. The flag **--dst** is an alias for this option.

-m, --match Specifies a match to use, that is, an extension module that tests for a specific property. The set of matches make up the condition under which a target is invoked. Matches are evaluated first to last as specified on the command line and work in short-circuit fashion, i.e. if one extension yields false, evaluation will stop.

-j, --jump *target* This specifies the target of the rule; i.e., what to do if the packet matches it. The target can be a user-defined chain (other than the one this rule is in), one of the special builtin targets which decide the fate of the packet immediately, or an extension (see **EXTENSIONS** below). If this option is omitted in a rule (and **-g** is not used), then matching the rule will have no effect on the packet's fate, but the counters on the rule will be incremented.

-g, --goto *chain* This specifies that the processing should continue in a user specified chain. Unlike the --jump option return will not continue processing in this chain but instead in the chain that called us via --jump.

[**!**] **-i, --in-interface** *name* of an interface via which a packet was received (only for packets entering the **INPUT**, **FORWARD** and **PREROUTING** chains). When the "!" argument is used before the interface name, the sense is inverted. If the interface name ends in a "+", then any interface which begins with this name will match. If this option is omitted, any interface name will match.

[**!**] **-o, --out-interface** *name* of an interface via which a packet is going to be sent (for packets entering the **FORWARD**, **OUTPUT** and **POSTROUTING** chains). When the "!" argument is used before the interface name, the sense is inverted. If the interface name ends in a "+", then any interface which begins with this name will match. If this option is omitted, any interface name will match.

[**!**] **-f, --fragment** This means that the rule only refers to second and further IPv4 fragments of fragmented packets. Since there is no way to tell the source or destination ports of such a packet (or ICMP type), such a packet will not match any rules which specify them. When the "!" argument precedes the "-f" flag, the rule will only match head fragments, or unfragmented packets. This option is IPv4 specific, it is not available in ip6tables.

-c, --set-counters *packets bytes* This enables the administrator to initialize the packet and byte counters of a rule (during **INSERT**, **APPEND**, **REPLACE** operations).

B.8 OTHER OPTIONS

The following additional options can be specified:

-v, --verbose output. This option makes the list command show the interface name, the rule options (if any), and the TOS masks. The packet and byte counters are also listed, with the suffix 'K', 'M' or 'G' for 1000, 1,000,000 and 1,000,000,000 multipliers respectively (but see the **-x** flag to change this). For appending, insertion, deletion and replacement, this causes detailed information on the rule or rules to be printed. **-v** may be specified multiple times to possibly emit more detailed debug statements.

-w, --wait [*seconds*] Wait for the xtables lock. To prevent multiple instances of the program from running concurrently, an attempt will be made to obtain an exclusive lock at launch. By default, the program will exit if the lock cannot be obtained. This option will make the program wait (indefinitely or for optional *seconds*) until the exclusive lock can be obtained.

-W, --wait-interval *microseconds* Interval to wait per each iteration. When running latency sensitive applications, waiting for the xtables lock for extended durations may not be acceptable. This option will make each iteration

take the amount of time specified. The default interval is 1 second. This option only works with **-w**.

-n, **--numeric** output. IP addresses and port numbers will be printed in numeric format. By default, the program will try to display them as host names, network names, or services (whenever applicable).

-x, **--exact** Expand numbers. Display the exact value of the packet and byte counters, instead of only the rounded number in K's (multiples of 1000) M's (multiples of 1000K) or G's (multiples of 1000M). This option is only relevant for the **-L** command.

--line-numbers When listing rules, add line numbers to the beginning of each rule, corresponding to that rule's position in the chain.

--modprobe=*command* When adding or inserting rules into a chain, use *command* to load any necessary modules (targets, match extensions, etc).

Appendix C LIST OF ACRONYMS AND ABBREVIATIONS

A list of acronyms and abbreviations is shown in Table C-1.

Table C-1. Acronyms and Abbreviations

Acronym/Abbreviation	Definition
AGD	Assurance Guidance Documents
AKA	Also Known As
cPP	Collaborative Protection Profile
DHE	Diffie–Hellman Key Exchange
EAL	Evaluation Assurance Levels
ECDH	Elliptic-Curve Diffie–Hellman
FIPS	Federal Information Processing Standard
FW	Firewall
GW	Gateway
HMAC	Hash-based Message Authentication Code
IA	Information Assurance
IT	Information Technology
NIAP	National Information Assurance Partnership
NTP	Network Time Protocol
PKI	Public Key Infrastructure
RMI	Remote Management Interface
RNG	Random Number Generator
SA	Security Administrator or Security Analysis
SAN	Storage Area Network
SAR	Security Assurance Requirement
SIEM	Security Information Event Management
SSH	Secure Shell Protocol
TA	Trust Anchor
TC	Trusted Channel
TCP	Transport Control Protocol
TSF	TOE Security Functions
TLS	Transport Layer Security
TOE	Target of Evaluation
TP	Trusted Path
TSF	TOE Security Functionality
TSS	TOE Summary Specification
VM	Virtual Machine
vND	Virtual Network Device