Extreme OS ONE Switching and Routing v22.2.2.0 Security Configuration Guide

GRUB Protection, ACLs, AAA, and Certificate Management

9039565-00 Rev AA
April 2026

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, see: https://www.extremenetworks.com/about-extreme-networks/company/legal/trademarks

Open Source Declarations

Some software files have been licensed under certain open source or third-party licenses. End-user license agreements and open source declarations can be found at: https://www.extremenetworks.com/support/policies/open-source-declaration/

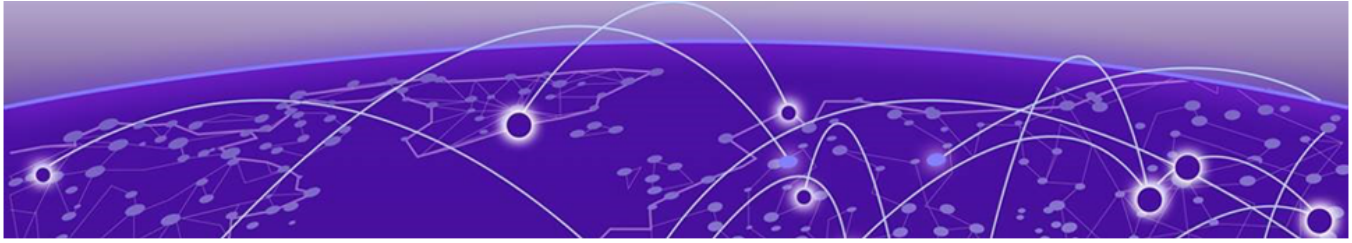
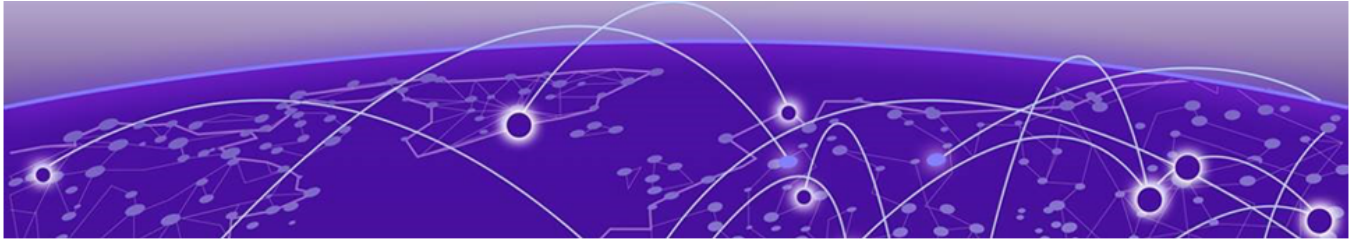


Table of Contents



Abstract

The *Extreme OS ONE SR Security Configuration Guide* version 22.2.2.0 provides advanced technical instructions for securing microservice-based network devices. Key features include GRUB boot loader protection using PBKDF password hashing, robust Access Control Lists (ACLs) with TCAM filtering and OpenConfig YANG model integration, and comprehensive AAA support for TACACS+, LDAP, and RADIUS with VRF-aware connectivity. The guide is intended for intermediate to advanced IT professionals responsible for secure switch deployment and management.

Preface

Read the following topics to learn about:

- The meanings of text formats used in this document.
- Where you can find additional information and help.
- How to reach us with questions and comments.

Text Conventions

Unless otherwise noted, information in this document applies to all supported environments for the products in question. Exceptions, like command keywords associated with a specific software version, are identified in the text.

When a feature, function, or operation pertains to a specific hardware product, the product name is used. When features, functions, and operations are the same across an entire product family, such as Extreme Networks switches, the product is referred to as *the switch*.

Table 1: Notes and warnings

Icon	Notice type	Alerts you to...
	Tip	Helpful tips and notices for using the product
	Note	Useful information or instructions
	Important	Important features or instructions
	Caution	Risk of personal injury, system damage, or loss of data
	Warning	Risk of severe personal injury

Table 2: Text

Convention	Description
`screen displays`	This typeface indicates command syntax, or represents information as it is displayed on the screen.
The words *enter* and *type*	When you see the word *enter* in this guide, you must type something, and then press the Return or Enter key. Do not press the Return or Enter key when an instruction simply says *type*.
Key names	Key names are written in boldface, for example **Ctrl** or **Esc**. If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example: Press **Ctrl**+**Alt**+**Del**
Words in italicized type	Italics emphasize a point or denote new terms at the place where they are defined in the text. Italics are also used when referring to publication titles.
NEW!	New information. In a PDF, this is searchable text.

Table 3: Command syntax

Convention	Description
bold text	Bold text indicates command names, keywords, and command options.
italic text	Italic text indicates variable content.
`[ ]`	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
`{ x | y | z }`	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, such as passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	In command examples, the backslash indicates a “soft” line break. When a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Documentation and Training

Find Extreme Networks product information at the following locations:

Current Product Documentation
Release Notes
Hardware and Software Compatibility for Extreme Networks products
Extreme Optics Compatibility
Other Resources such as articles, white papers, and case studies

Open Source Declarations

Some software files have been licensed under certain open source licenses. Information is available on the Open Source Declaration page.

Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For details, visit the Extreme Networks Training page.

Help and Support

If you require assistance, contact Extreme Networks using one of the following methods:

Extreme Portal

Search the GTAC (Global Technical Assistance Center) knowledge base; manage support cases and service contracts; download software; and obtain product licensing, training, and certifications.

The Hub

A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.

Call GTAC

For immediate support: (800) 998 2408 (toll-free in U.S. and Canada) or 1 (408) 579 2800. For the support phone number in your country, visit www.extremenetworks.com/support/contact.

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number, or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any actions already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

Subscribe to Product Announcements

You can subscribe to email notifications for product and software release announcements, Field Notices, and Vulnerability Notices.

1. Go to The Hub.
2. In the list of categories, expand the **Product Announcements** list.
3. Select a product for which you would like to receive notifications.
4. Select **Subscribe**.
5. To select additional products, return to the **Product Announcements** list and repeat steps 3 and 4.

You can modify your product selections or unsubscribe at any time.

Send Feedback

The User Enablement team at Extreme Networks has made every effort to ensure that this document is accurate, complete, and easy to use. We strive to improve our documentation to help you in your work, so we want to hear from you. We welcome all feedback, but we especially want to know about:

- Content errors, or confusing or conflicting information.
- Improvements that would help you find relevant information.
- Broken links or usability issues.

To send feedback, email us at Product-Documentation@extremenetworks.com.

Provide as much detail as possible including the publication title, topic heading, and page number (if applicable), along with your comments and suggestions for improvement.

About This Document

What's New in This Document

The following table describes information added to this guide for Extreme OS ONE Switching and Routing, release 22.2.2.0.

Feature	Description	Link
Policy-Based Routing	Updated topic "Command-Line Interface (CLI)" with the commands to configure ACEs and ACLs.	Command-Line Interface (CLI) on page 18
ACL QoS and Mirroring	New topic "ACL QoS and Mirroring" describes how to apply QoS settings via ACLs on Layer 2 or Layer 3 ingress security and configure ACL mirroring.	ACL QoS and Mirroring on page 43
Management Interface ACL	New topic "Connection Limit on Management Interface" describes configuring connection limits on management interfaces for IPv4 and IPv6 ACLs.	Connection Limit on Management Interface on page 39
OOBM and Breakout Ports	Updated topic "Attachment Points" with the details on out of band management (OOBM) and breakout ports.	Attachment Points on page 21
Password Reuse Policy	New topic "Set the Password Reuse Policy" describes how to configure the number of prior passwords against which the system checks a new password.	Set the Password Reuse Policy on page 48

Feature	Description	Link
Password Entropy	Updated topic "Password Requirement Attributes and Default Password Strength" now describes how to specify the permitted maximum numbers of consecutive characters and repetitive characters in new passwords.	Password Requirement Attributes and Default Password Strength on page 49
ICMP Rate Limiting on the Management Interface	New topic "ICMP Rate Limiting on the Management Interface" describes how to control the number of ICMP packets that the management interface processes within a defined threshold. This prevents abuse while maintaining functionality for diagnostics.	ICMP Rate Limiting on the Management Interface on page 83
Mutual Authentication	New chapter "Mutual Authentication" describes how to configure mutual TLS (mTLS) for the Remote Procedure Calls (gRPC) server as well as for critical services such as *LDAP*, *RADIUS*, and remote system logging (*SYSLOG*). It also describes how to configure mTLS for the HTTPS client that you use for system firmware operations.	Mutual Authentication on page 97

For more information, see the *Extreme OS ONE SR Release Notes*.

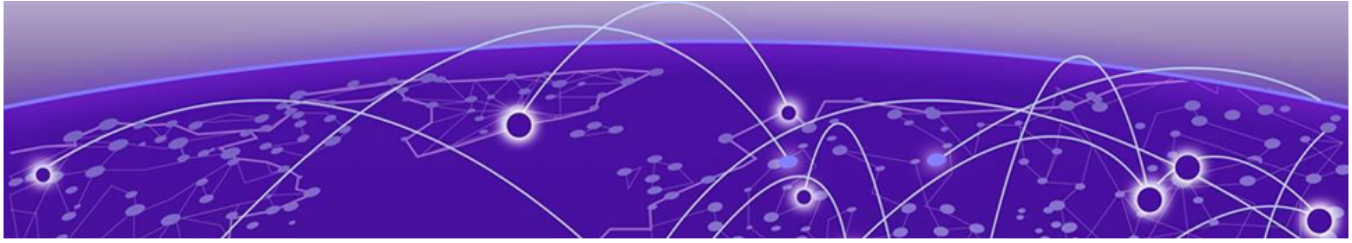
Supported Platforms

Extreme OS ONE Switching and Routing 22.2.1.0 and later releases support Extreme 8520, Extreme 8720, Extreme 8730, and Extreme 8820 hardware platforms.

Note

Although many software and hardware configurations are tested and supported for this release, documenting all possible configurations and scenarios is beyond this document's scope.

For information about other releases, see the documentation for those releases.



Securing GRUB

Securing GRUB Boot Loader on page 14

Securing GRUB Boot Loader

GRUB's boot loader interface is accessible to anyone with console access. Users can edit boot menu entries, add or delete entries, or access the GRUB prompt. To secure this feature, the system provides a default username and password to protect the GRUB menu. Extreme OS ONE also provides a CLI and API to modify these credentials and restrict access to the boot loader menu.

The feature is available on all the platforms supported by Extreme OS ONE and enables you to perform the following tasks:

- Include a default Grub username and password to protect the Grub boot loader menu.
- Require users to change the default credentials (username and/or password) when logging in to the CLI, with the option to reuse the admin password.
- Change the default username or password using the CLI or gNMI.
- Configure GRUB credentials using a ZTP configuration file during initial provisioning.
- Generate a warning message during each boot if the default GRUB credentials remain unchanged.
- Require authentication for any action other than selecting the default boot option, including booting other entries, editing entries, or accessing the GRUB command-line interface.

Configure GRUB Boot Loader Credentials using CLI

Configure GRUB credentials to protect the boot loader by running the following commands as an admin user:

```
device# configure terminal
device(config)# system
device(config-system)# grub
device(config-system-grub)# username root password <password>
```

The username must start with an alphanumeric or underscore character and can contain only alphanumeric, underscore, or period characters.

The plain-text password must satisfy the password strength requirements defined in password-attributes (under aaa authentication password-attributes).

Key points:

- The system supports only one user for GRUB protection, with the default username root.
- You can change credentials using the CLI or gNMI.
- The system prompts users to change the default password at first login.
- You can set the GRUB password independently or synchronize it with the admin password.
- You can provide passwords in plain text or as a generated hash.
- The system stores the boot loader configuration, including the username and password hash, in the internal configuration database (CDB) and is also appended to the GRand Unified Bootloader (GRUB) configuration file on the device.

  GRUB Authentication

  - GRUB requires authentication for non-default boot entries, editing, and command-line access.
  - The default boot entry (Open Network Linux) does not require authentication to allow system boot.
  - The same credentials protect all GRUB menu entries, including ONIE and diagnostic options.

  Note
  Separate credentials for different boot menu entries are not supported.

Configure GRUB Boot Loader Credentials using the gNMI Command

Use the path `system/grub/config/username`.

Configure GRUB Boot Loader Credentials using the Copy Config Command

Using Default Configuration Copy

When you run `copy default-config running-config` or perform a factory reset, the GRUB credentials in grub.cfg revert to default values before the device reboots. After the reboot, use the default credentials to access the GRUB menu, except for the default boot entry.

Using User Configuration Copy

If you copy a user configuration, the modified GRUB username and password will be applied to grub.cfg. After the reboot, you'll need to use the new credentials for non-default boot menu entries.

GRUB Password Protection Configuration

- During zero touch provisioning (ZTP): You can modify GRUB credentials using the **username** (GRUB system configuration). For details, see the *Extreme OS ONE SR Command Reference Guide*.
- During downgrades: When downgrading to a lower version image that doesn't support this feature, password entries in grub.cfg will be removed, and GRUB menu entries won't require password authentication.
- During fresh install: A fresh install of Extreme OS ONE using ONIE re-creates the default username and password in the configuration database and `/mnt/onl/boot/grub.cfg`.g.

Special Boot Modes

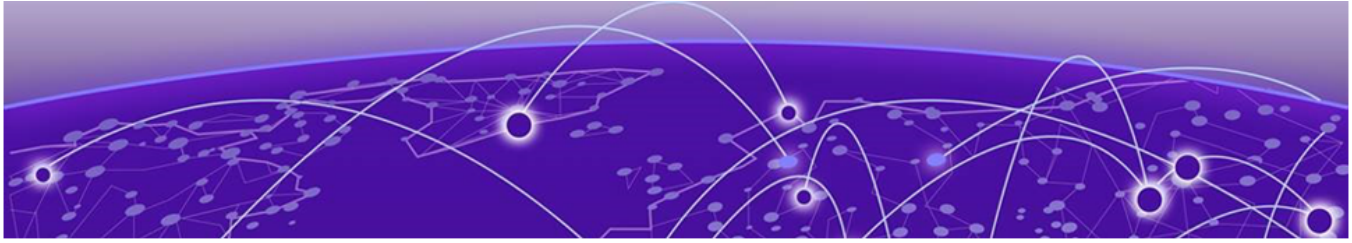
- Diag boot: The internal `run diag` CLI command allows the system to boot into diagnostics mode without requiring a GRUB password. After the diagnostic run completes, password protection is restored. The username and password remain unchanged.
- Full install: The `fullinstall` CLI command allows the system to boot into ONIE mode without requiring a GRUB password. After a full install, the device installs a new version of Extreme OS ONE with default GRUB credentials. installation with default username/password for GRUB.

Lost Password

The device boots into Extreme OS ONE without requiring a GRUB password. If the username or password is lost, change it using the CLI. If you lose or forget the username or password, you can reset them by using the **username** (GRUB system configuration) command as described previously.

Security

GRUB uses a password hashing algorithm based on the Password-Based Key Derivation Function (PBKDF), as defined in RFC 2898, to ensure secure password storage.



ACLs

Access Control List (ACL) Overview

An Access Control List (ACL) is a set of rules, called Access Control Entries (ACEs), that you can attach to an attachment point. An ACL takes effect only when you attach it to an attachment point.

Key Characteristics

- Sequence ID: Each ACE has a sequence ID that determines priority. Lower sequence IDs have higher priority.
- TCAM: ACLs use ternary content-addressable memory (TCAM) to filter packets. Each ACE is programmed as a TCAM entry.
- Matching Logic: The system matches all entries simultaneously and applies the action from the highest-priority match (lowest sequence ID).

For information about how multiple ACL features interact, see ACL Evaluation and Precedence on page 26.

Types of ACLs

- MAC ACL: Matches MAC (link layer) fields in the packet header, with optional metadata.
- IPv4 ACL: Matches IPv4 (network layer) fields, with optional metadata and transport layer (TCP/UDP) fields.

- IPv6 ACL: Matches IPv6 (network layer) fields, with optional metadata and transport layer fields.

Note
Metadata refers to data derived from the packet by the switch, such as port information, routability, and bridge domain.

Command-Line Interface (CLI)

The CLI is the primary interface for configuring networking devices. Use the CLI to create, modify, and display ACL configurations and statistics.

Use the following CLI commands:

1. Create or delete an ACL.
```
device(config)# [no] (ipv4 | ipv6 | mac) access-list <ACL_NAME>
```
2. Configure access control entries (ACEs).
   - Configure an IPv4 ACE.
```
device(config-ipv6-acl)# [seq <1-65535>] permit  (<1-254> | ipv4 | icmp | igmp | esp| udp | tcp
[ack] [cwr] [ece] [fin] [push] [reset] [sync] [urg]) (<SRC_IP> <MASK> | <SRC_IP/PREFIX> | any)
(<DST_IP> <MASK> | <DST_IP/PREFIX> | any)  [dport <1-65535>] [sport <1-65535>]  [dscp <0-63>]
[vlan <0-4095>] [count] [mirror <NAME>]  [force-tc <0-7>] [force-dscp <0-63>] [force-dp <0-2>]
[force-pcp <0-7>] [connlimit <1-65535>]

device(config-ipv6-acl)# [seq <1-65535>]  deny (<1-254> | ipv4 | icmp | igmp | esp| udp | tcp
[ack] [cwr] [ece] [fin] [push] [reset] [sync] [urg]) (<SRC_IP> <MASK> | <SRC_IP/PREFIX> | any)
(<DST_IP> <MASK> | <DST_IP/PREFIX> | any)  [dport <1-65535>] [sport <1-65535>]  [dscp <0-63>]
[vlan <0-4095>] [count] [connlimit <1-65535>]
```
   - Configure an IPv6 ACE.
```
device(config-ipv6-acl)# [seq <1-65535>] permit (<1-254> | ipv6 | icmpv6 | esp | udp | tcp
[ack] [cwr] [ece] [fin] [push] [reset] [sync] [urg]) (<SRC_IP> <MASK> | <SRC_IP/PREFIX> | any)
(<DST_IP> <MASK> | <DST_IP/PREFIX> | any) [dport <1-65535>] [sport <1-65535>] [dscp <0-63>]
[vlan <0-4095>] [count] [mirror <NAME>]  [force-tc <0-7>] [force-dscp <0-63>] [force-dp <0-2>]
[force-pcp <0-7>] [connlimit <1-65535>]

device(config-ipv6-acl)# [seq <1-65535>] deny (<1-254> | ipv6 | icmpv6 | esp | udp | tcp
[ack] [cwr] [ece] [fin] [push] [reset] [sync] [urg]) (<SRC_IP> <MASK> | <SRC_IP/PREFIX> | any)
(<DST_IP> <MASK> | <DST_IP/PREFIX> | any) [dport <1-65535>] [sport <1-65535>] [dscp <0-63>]
[vlan <0-4095>] [count]  [connlimit <1-65535>]
```
   - Configure MAC ACE.
```
device(config-mac-acl)# [seq <1-65535>] permit (<SRC_ADDRESS> <MASK> | any ) (<DST_ADDRESS>
<MASK> | any ) [etype <ETHTYPE>] [vlan <0-4095>] [pcp <1-7>] [count] [mirror <NAME>]  [force-tc
<0-7>] [force-pcp<0-7>] [force-dscp <0-63>] [force-dp <0-2>]

device(config-mac-acl)# [seq <1-65535>] deny (<SRC_ADDRESS> <MASK> | any ) (<DST_ADDRESS>
<MASK> | any ) [etype <ETHTYPE>] [vlan <0-4095>] [pcp <1-7>] [count]
```
3. Show Commands.
   - To view the configuration.
```
device# show running-config  [(ipv4 | ipv6 | mac)] access-list (all | <NAME>)
```
   - To view the state and statistics
```
device# show [(ipv4 | ipv6 | mac)] access-list (all | <NAME>)
```

For details about these commands (including the **force-tc**, **force-dscp**, **force-pcp**, and **force-dp** options for ACL QoS support on Layer 2 and Layer 3 ingress security groups), see the *Extreme OS ONE SR Command Reference Guide*.

YANG Model for ACL Configuration

The YANG model defines the structure used to configure ACLs, including ACL sets, ACEs, and actions.

The OpenConfig ACL YANG model is used for ACL configuration, with some additional fields augmented to the main tree. The /acl/acl-sets branch stores ACLs.

Key Components

- ACL sets: The `/acl/acl-sets` branch stores ACLs, indexed by name and type.
- ACEs: The `acl-entry*` branch stores entries, indexed by sequence ID.

YANG Tree

The following YANG tree shows the ACL structure:

```
+--rw acl
     +--rw config
     |  +--rw extr-acl-ext:count?   boolean
     +--ro state
     |  +--ro extr-acl-ext:count?   boolean
     +--rw acl-sets
     |  +--rw acl-set* [name type]
     |     +--rw name           -> ../config/name
     |     +--rw type           -> ../config/type
     |     +--rw config
     |     |  +--rw name?   string
     |     |  +--rw type?   identityref
     |     +--ro state
     |     |  +--ro name?   string
     |     |  +--ro type?   identityref
     |     +--rw acl-entries
     |        +--rw acl-entry* [sequence-id]
     |           +--rw sequence-id                  -> ../config/sequence-id
     |           +--rw config
     |           |  +--rw sequence-id?   uint32
     |           +--ro state
     |           |  +--ro sequence-id?       uint32
     |           |  +--ro matched-packets?   oc-yang:counter64
     |           |  +--ro matched-octets?    oc-yang:counter64
     |           +--rw actions
     |           |  +--rw config
     |           |  |  +--rw forwarding-action          identityref
     |           |  |  +--rw extr-acl-ext:count?        boolean
     |           |  |  +--rw extr-acl-ext:mirror?       string
     |           |  |  +--rw extr-acl-ext:force-tc?     uint8
     |           |  |  +--rw extr-acl-ext:force-pcp?    uint8
     |           |  |  +--rw extr-acl-ext:force-dscp?   uint8
     |           |  |  +--rw extr-acl-ext:force-dp?     uint8
     |           |  +--ro state
     |           |     +--ro forwarding-action          identityref
     |           |     +--ro extr-acl-ext:count?        boolean
     |           |     +--ro extr-acl-ext:mirror?       string
```

```
     |           |     +--ro extr-acl-ext:force-tc?     uint8
     |           |     +--ro extr-acl-ext:force-pcp?    uint8
     |           |     +--ro extr-acl-ext:force-dscp?   uint8
     |           |     +--ro extr-acl-ext:force-dp?     uint8
     |           +--rw extr-acl-mac-ext:acl-mac
     |           |  +--rw extr-acl-mac-ext:config
     |           |  |  +--rw extr-acl-mac-ext:source-mac?             oc-yang:mac-address
     |           |  |  +--rw extr-acl-mac-ext:source-mac-mask?        oc-yang:mac-address
     |           |  |  +--rw extr-acl-mac-ext:destination-mac?        oc-yang:mac-address
     |           |  |  +--rw extr-acl-mac-ext:destination-mac-mask?   oc-yang:mac-address
     |           |  |  +--rw extr-acl-mac-ext:pcp?                    uint8
     |           |  |  +--rw extr-acl-mac-ext:ethertype?              oc-pkt-match-types:ethertype-type
     |           |  |  +--rw extr-acl-mac-ext:vlan-tag?               uint16
     |           |  +--ro extr-acl-mac-ext:state
     |           |     +--ro extr-acl-mac-ext:source-mac?             oc-yang:mac-address
     |           |     +--ro extr-acl-mac-ext:source-mac-mask?        oc-yang:mac-address
     |           |     +--ro extr-acl-mac-ext:destination-mac?        oc-yang:mac-address
     |           |     +--ro extr-acl-mac-ext:destination-mac-mask?   oc-yang:mac-address
     |           |     +--ro extr-acl-mac-ext:pcp?                    uint8
     |           |     +--ro extr-acl-mac-ext:ethertype?              oc-pkt-match-types:ethertype-type
     |           |     +--ro extr-acl-mac-ext:vlan-tag?               uint16
     |           +--rw extr-acl-ipv4-ext:acl-ipv4
     |           |  +--rw extr-acl-ipv4-ext:config
     |           |  |  +--rw extr-acl-ipv4-ext:source-ipv4?             oc-inet:ipv4-address
     |           |  |  +--rw extr-acl-ipv4-ext:source-ipv4-mask?        oc-inet:ipv4-address
     |           |  |  +--rw extr-acl-ipv4-ext:destination-ipv4?        oc-inet:ipv4-address
     |           |  |  +--rw extr-acl-ipv4-ext:destination-ipv4-mask?   oc-inet:ipv4-address
     |           |  |  +--rw extr-acl-ipv4-ext:dscp?                    oc-inet:dscp
     |           |  |  +--rw extr-acl-ipv4-ext:protocol?                oc-pkt-match-types:ip-protocol-
type
     |           |  |  +--rw extr-acl-ipv4-ext:vlan-tag?                uint16
     |           |  |  +--rw extr-acl-ipv4-ext:source-port?             oc-pkt-match-types:port-num-
range
     |           |  |  +--rw extr-acl-ipv4-ext:destination-port?        oc-pkt-match-types:port-num-
range
     |           |  |  +--rw extr-acl-ipv4-ext:tcp-flags*               identityref
     |           |  +--ro extr-acl-ipv4-ext:state
     |           |     +--ro extr-acl-ipv4-ext:source-ipv4?             oc-inet:ipv4-address
     |           |     +--ro extr-acl-ipv4-ext:source-ipv4-mask?        oc-inet:ipv4-address
     |           |     +--ro extr-acl-ipv4-ext:destination-ipv4?        oc-inet:ipv4-address
     |           |     +--ro extr-acl-ipv4-ext:destination-ipv4-mask?   oc-inet:ipv4-address
     |           |     +--ro extr-acl-ipv4-ext:dscp?                    oc-inet:dscp
     |           |     +--ro extr-acl-ipv4-ext:protocol?                oc-pkt-match-types:ip-protocol-
type
     |           |     +--ro extr-acl-ipv4-ext:vlan-tag?                uint16
     |           |     +--ro extr-acl-ipv4-ext:source-port?             oc-pkt-match-types:port-num-
range
     |           |     +--ro extr-acl-ipv4-ext:destination-port?        oc-pkt-match-types:port-num-
range
     |           |     +--ro extr-acl-ipv4-ext:tcp-flags*               identityref
     |           +--rw extr-acl-ipv6-ext:acl-ipv6
     |              +--rw extr-acl-ipv6-ext:config
     |              |  +--rw extr-acl-ipv6-ext:source-ipv6?             oc-inet:ipv6-address
     |              |  +--rw extr-acl-ipv6-ext:source-ipv6-mask?        oc-inet:ipv6-address
     |              |  +--rw extr-acl-ipv6-ext:destination-ipv6?        oc-inet:ipv6-address
     |              |  +--rw extr-acl-ipv6-ext:destination-ipv6-mask?   oc-inet:ipv6-address
     |              |  +--rw extr-acl-ipv6-ext:dscp?                    oc-inet:dscp
     |              |  +--rw extr-acl-ipv6-ext:protocol?                oc-pkt-match-types:ip-protocol-
type
     |              |  +--rw extr-acl-ipv6-ext:vlan-tag?                uint16
     |              |  +--rw extr-acl-ipv6-ext:source-port?             oc-pkt-match-types:port-num-
range
     |              |  +--rw extr-acl-ipv6-ext:destination-port?        oc-pkt-match-types:port-num-
range
```

```
     |              |  +--rw extr-acl-ipv6-ext:tcp-flags*              identityref
     |              +--ro extr-acl-ipv6-ext:state
     |                 +--ro extr-acl-ipv6-ext:source-ipv6?            oc-inet:ipv6-address
     |                 +--ro extr-acl-ipv6-ext:source-ipv6-mask?       oc-inet:ipv6-address
     |                 +--ro extr-acl-ipv6-ext:destination-ipv6?       oc-inet:ipv6-address
     |                 +--ro extr-acl-ipv6-ext:destination-ipv6-mask?  oc-inet:ipv6-address
     |                 +--ro extr-acl-ipv6-ext:dscp?                   oc-inet:dscp
     |                 +--ro extr-acl-ipv6-ext:protocol?               oc-pkt-match-types:ip-protocol-
type
     |                 +--ro extr-acl-ipv6-ext:vlan-tag?               uint16
     |                 +--ro extr-acl-ipv6-ext:source-port?            oc-pkt-match-types:port-num-
range
     |                 +--ro extr-acl-ipv6-ext:destination-port?       oc-pkt-match-types:port-num-
range
     |                 +--ro extr-acl-ipv6-ext:tcp-flags*              identityref
```

Statistics

The system collects statistics for each ACE in hardware and updates the matched-packets and matched-octets fields in the YANG model.

ACL Configuration Validators

The API-GW validates ACL configurations and checks for consistency. If the configuration is invalid or conflicting, the system returns an error.

Security ACL validations:

- seq ID with different match criteria: Seq ID Exists
- seq ID with different action: Conflicting entry
- seq ID with same match and action: Duplicate entry

API-GW is the central point for configuration validation, handling inputs from CLI, EVM, and GNMI.

Attachment Points

An attachment point defines where an ACL is applied. The following attachment points are supported:

- Physical Interface: Apply an ACL to filter traffic on a physical port (for example, ethernet 0/1).
- LAG (Link Aggregation Group): Apply an ACL to filter traffic on a link aggregation group.

Note
ACLs cannot be attached to LAG group members.

- VE (Virtual Ethernet): Apply an ACL to filter traffic in a bridge domain.

  Note
  ACLs attached to physical ports and LAGs have higher priority than ACLs on VEs.

- Control Plane: Apply an ACL to filter traffic sent to the CPU (RACL).
- OOBM Port: Apply ACLs using Linux kernel TC filters on the host OS.
- Breakout Port: Treat breakout ports as physical ports for ACL processing.
  - When you convert a physical port to a breakout port, the system removes ACLs on the original port.
  - When you convert a breakout port back to a physical port, the system removes ACLs on breakout ports.

If an ACL is attached to an attachment point, the rules will be programmed to hardware. This is regardless of whether the attachment point is admin/operational up or down.

If you delete a LAG or VE port, the system removes only the ACL attachment configuration. The ACL configuration remains unchanged. If you recreate the port, reattach the ACL. You do not need to recreate the ACL.

If a physical port is converted to a breakout port or vice versa, only the ACL attachment configuration under the deleted physical port is deleted, and the ACL configuration itself remains intact. If you re-create the port, you need to re-attach the ACL to it (and do not need to re-create the ACL configuration).

Management Interface ACLs

You can attach a Security ACL to the management interface to restrict inbound access to management services. Management interface ACLs differ from interface ACLs in supported match fields and actions, and they support optional connection limiting.

For details, see Security ACLs on the Management Interface on page 37.

Attachment Direction

The attachment direction defines whether the ACL filters incoming or outgoing traffic. By specifying the attachment direction, you can control whether the ACL filters incoming or outgoing packets on a particular attachment point. There are two directions:

- Ingress: Filters packets entering the switch
- Egress: Filters packets leaving the switch

Security ACL

A security ACL controls access to resources by matching packet fields and applying actions.

Key Components

- Rules combine match conditions and actions.
- Conditions match packet header fields or metadata.
- Actions include:
  - Permit or deny
  - QoS actions (traffic class, COS, DSCP, drop precedence)
  - Mirroring

How it Works

To configure a security ACL:

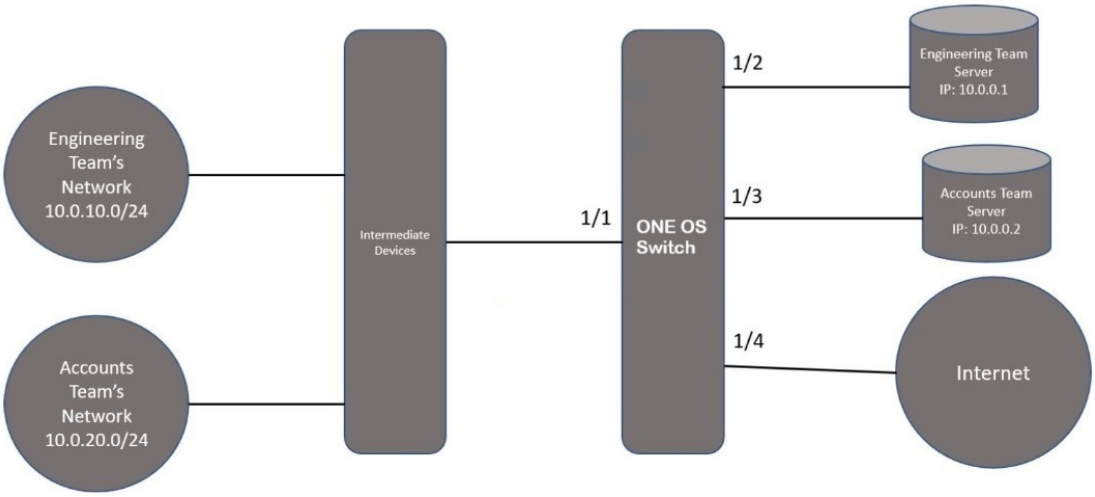
1. Create an ACL: Define an ACL with specific rules to filter traffic.
2. Attach the ACL: Attach the ACL to an attachment point (for example, a physical interface) in the required direction (ingress or egress).

Example Use Case

Restrict access between two teams (Engineering and Accounts) while allowing internet access to all

- Engineering team: 192.0.2.0/24
- Accounts team: 192.0.3.0/24

The goal is to prevent the engineering team from accessing the accounts team's server and vice versa while providing internet access to all teams. Use a security ACL to achieve the desired traffic filtering and access control.



The following examples show how ACLs filter traffic based on defined rules.

Ingress ACL example

Create an ACL and attach it to the ingress direction on interface 1/1:

```
ipv4 access-list ipAcl
  seq 10 permit ipv4 192.0.2.0/24 192.0.2.1
  seq 20 deny ipv4 any 192.0.2.1
  seq 30 permit ipv4 198.51.100.0/24 198.51.100.2
  seq 40 deny ipv4 any 198.51.100.2
```

```
  seq 50 permit ipv4 any any
interface ethernet 1/1
  ipv4 access-list ipAcl in
```

Note

ACLs on this platform have an implicit deny ipv4 any any at the end of every ACL. Any traffic not explicitly matched by a preceding rule is dropped. In this example, seq 50 permit ipv4 any any is required to allow internet access for both teams. Without it, all traffic not destined for the two servers would be silently dropped by the implicit deny.

Egress ACL example

Create separate ACLs and attach them to interfaces 1/2 and 1/3 in the egress direction:

```
ipv4 access-list ipAclEngineering
  seq 10 permit ipv4 192.0.2.0/24 any
ipv4 access-list ipAclAccounts
  seq 10 permit ipv4 198.51.100.0/24 any
interface ethernet 1/2
  ipv4 access-list ipAclEngineering out
interface ethernet 1/3
  ipv4 access-list ipAclAccounts out
```

CLI Configuration Commands

Use the following command to attach an ACL:

```
device(config)# interface (ethernet | ve | port-channel) <INTERFACE_NAME>
device(config-intf-<type>)# [no] (ipv4 | ipv6 | mac) access-list <ACL_NAME> (in | out)
```

To verify the configuration, use the following show commands:

```
device# show running-config interface (ethernet | ve | port-channel) <INTERFACE_NAME>
device# show interface (ethernet | ve | port-channel) <INTERFACE_NAME>
```

YANG Model for ACL Attachments

The openconfig-acl YANG model is used for ACL attachments to an interface. The `/acl/interfaces` branch is used to attach an ACL to an interface.

Key Components

1. The interface table is indexed by interface ID (for example, "ethernet 0/1", "ve 10", "port-channel 1").
2. The ingress ACL sets attach ACLs to an interface for ingress traffic.
3. The egress ACL sets attach ACLs to an interface for egress traffic.

YANG Tree

The YANG tree structure is as follows:

```
+-rw acl
 +-rw interfaces
    +-rw interface* [id]
       +-rw id
```

```
      +-rw config
      |  +-rw id
      +-ro state
      |  +-ro id
      +-rw ingress-acl-sets
      |  +-rw ingress-acl-set* [set-name type]
      |     +-rw set-name
      |     +-rw type
      |     +-rw config
      |     |  +-rw set-name
      |     |  +-rw type
      |     +-ro state
      |     |  +-ro set-name
      |     |  +-ro type
      |     +-ro acl-entries
      |        +-ro acl-entry* [sequence-id]
      |           +-ro sequence-id
      |           +-ro state
      |              +-ro sequence-id
      +-rw egress-acl-sets
         +-rw egress-acl-set* [set-name type]
            +-rw set-name
            +-rw type
            +-rw config
            |  +-rw set-name
            |  +-rw type
            +-ro state
            |  +-ro set-name
            |  +-ro type
            +-ro acl-entries
               +-ro acl-entry* [sequence-id]
                  +-ro sequence-id
                  +-ro state
                     +-ro sequence-id
```

Configuration Validators

If an ACL of the same type is already configured on an attachment point, attaching another ACL of the same type will result in an error.

Receive ACL (RACL)

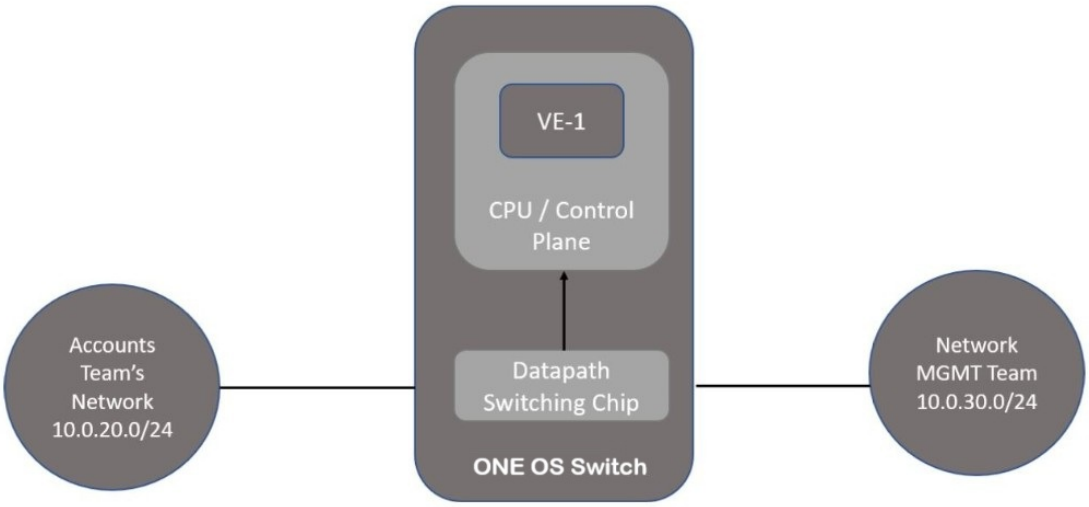
A receive ACL (RACL), also called a control plane ACL, filters traffic destined for the device CPU. It's designed to protect the device from unauthorized access and attacks.

Key Features

1. Filters CPU-bound traffic. RACL filters traffic destined to the device's IP interfaces, such as FTP, Telnet, and other management traffic.
2. Applies globally to the device. RACLs are applied globally to the device, not specific to any interface, LAG, VE, or VRF.
3. Supports only IP ACLs. RACLs only support IP ACLs, not MAC ACLs.

Example Use Case: To restrict access to the control plane for specific networks or teams, such as allowing only the Network Management team to access the switch's

IP interface, define an IP ACL with permit or deny rules, and apply it to the control plane using the control-plane command.



Configuration Example

To allow only the Network Management team to access the control plane, create an ACL and apply it to the control plane:

```
ipv4 access-list ipAcl
  seq 10 permit ipv4 203.0.113.0/24 any
  seq 20 deny ip any any
control-plane
  ipv4 access-list ipAcl in
```

YANG and CLI

The same YANG model used for security ACLs is used for RACLs.

To attach an ACL to the control plane, use the following CLI:

```
device(config)# control-plane
device(config-control-plane)# [no] (ipv4 | ipv6) access-list <acl-name>
```

Configuration Validators

Attaching an ACL to the control plane when the same type is already configured will cause an error.

ACL Evaluation and Precedence

A packet can match multiple ACL features on the device. The system evaluates these features based on defined precedence rules.

Sequence ID Priority

Within a single ACL, entries are evaluated based on sequence ID. Lower sequence IDs have higher priority. If multiple entries match, the system applies the action from the entry with the lowest sequence ID.

Cross-Feature Evaluation

When multiple ACL features apply, a deny action takes precedence over a permit action.

Control-plane protection mechanisms, such as RACLs, are evaluated before data-plane Security ACLs.

Drop Precedence

If any applicable ACL feature determines that a packet must be dropped, the packet is dropped, even if another ACL feature permits the traffic.

Implicit Deny

All ACLs have an implicit deny ipv4 any any (or deny ipv6 any any for IPv6 ACLs) as the final rule. This implicit deny is not visible in the ACL configuration or statistics output, but it is always in effect. If a packet does not match any explicitly configured ACE in the ACL, it is dropped by this implicit deny.

Note
To allow traffic that would otherwise be dropped by the implicit deny, add an explicit permit ipv4 any any (or permit ipv6 any any) as the final ACE in your ACL. Ensure this rule is assigned the highest sequence ID in the ACL so it is evaluated last.

ACL Platform Support

The following sections describe the supported match fields and actions for each platform.

Extreme 8730 Platforms

The following tables summarize the qualifiers and attachment points supported for various features on the Extreme 8730 platform.

Table 4: Ingress security ACL

Type	Qualifier	Action	Scale	Note
IPv4	s-ip, d-ip, ip-protocol(tcp/udp/icmp/igmp/esp/customProtocolNumber), TcpFlags, L4SPort, L4DPort, vlan, dscp	Permit, Deny, Count, force-tc, force-pcp,force-dscp, force-dp, mirror	2047	TCAM Depth: The TCAM has a depth of 2048 entries. However, the available user ACL scale is less due to reserved system entries. This scale may decrease further with additional feature implementations.
IPv6	s-ip, d-ip, ip-protocol(tcp/udp/esp/icmpv6/customProtocolNumber), TcpFlags, L4SPort, L4DPort, vlan, dscp			
MAC	s-mac, d-mac, vlan, ethtype, cos			

Table 5: Egress security ACL

Type	Qualifier	Action	Scale	Note
IPv4	s-ip, d-ip, ip-protocol(tcp/udp/icmp/igmp/esp/customProtocolNumber), TcpFlags, L4SPort, L4DPort, dscp	Permit, Deny, Count	2047	TCAM Depth: The TCAM has a depth of 2048 entries. However, the available user ACL scale is less due to reserved system entries. This scale may decrease further with additional feature implementations.
IPv6	s-ip, d-ip, ip-protocol(tcp/udp/esp/icmpv6/customProtocolNumber), TcpFlags,			

Table 5: Egress security ACL (continued)

Type	Qualifier	Action	Scale	Note
	L4SPort, L4DPort, dscp			
MAC	s-mac, ethtype, cos			

Table 6: RACL

Type	Qualifier	Action	Scale
IPv4	s-ip, d-ip, ip-protocol(tcp/udp/icmp/igmp/esp/customProtocolNumber), TcpFlags, L4SPort, L4DPort, dscp	Permit, Deny, Count	2048
IPv6	s-ip, ip-protocol(tcp/udp/esp/icmpv6/customProtocolNumber), TcpFlags, L4SPort, L4DPort, dscp		

Table 7: Ingress and egress security ACL attachment points

Interface type	ACL type	Ingress support	Egress support
Physical	IPv4	Yes	Yes
	IPv6		
	MAC		
LAG	IPv4		No
	IPv6		
	MAC		
VE	IPv4		No
	IPv6		
	MAC		
Management interface	IPv4 / IPv6		No

Table 8: RACL attachment points

Interface type	ACL type	Ingress support	Egress support
	IPv4	Yes	n/a

Table 8: RACL attachment points (continued)

Interface type	ACL type	Ingress support	Egress support
Control plane	IPv6		

Extreme 8820 Platform

The following tables summarize the qualifiers and attachment points supported for various features on the Extreme 8820 platform. On these platforms, the TCP flag field is 6 bit and hence cannot support the CWR and ECN bits.

Table 9: Ingress security ACL

Type	Qualifier	Action	Scale	Note
IPv4	s-ip, d-ip, ip-protocol(tcp/udp/icmp/igmp/esp/customProtocolNumber),TcpFlags, L4SPort, L4DPort, dscp, vlan	Permit, Deny, Count, force-tc, force-pcp, force-dscp, force-dp, mirror	4096	TCAM Depth: The TCAM has a depth of 2048 entries. However, the available user ACL scale is less due to reserved system entries. This scale may decrease further with additional feature implementations.
IPv6	s-ip, ip-protocol(tcp/udp/esp/icmpv6/customProtocolNumber), TcpFlags, L4SPort, L4DPort, dscp, vlan			
MAC	s-mac, d-mac, vlan, ethtype, cos	Permit, Deny, Count, force-tc, force-pcp,force-dp, mirror	2044	

Table 10: Egress security ACL

Type	Qualifier	Action	Scale	Note
IPv4	s-ip, d-ip, ip-protocol(tcp/udp/icmp/igmp/esp/customProtocolNumber), TcpFlags, L4SPort, L4DPort, vlan, dscp	Permit, Deny, Count, Mirror	2047	TCAM Depth: The TCAM has a depth of 2048 entries. However, the available user ACL scale is less due to reserved system entries. This scale may decrease further with additional feature implementations.
IPv6	s-ip, d-ip, ip-protocol(tcp/udp/esp/icmpv6/customProtocolNumber), TcpFlags, L4SPort,			

Table 10: Egress security ACL (continued)

Type	Qualifier	Action	Scale	Note
	L4DPort, vlan, dscp			

Table 11: RACL

Type	Qualifier	Action	Scale
IPv4	s-ip, d-ip, ip-protocol(tcp/udp/icmp/igmp/esp/customProtocolNumber),TcpFlags, L4SPort, L4DPort, dscp	Deny, Permit, Count.	2048
IPv6	s-ip, ip-protocol(tcp/udp/esp/icmpv6/customProtocolNumber), TcpFlags, L4SPort, L4DPort, dscp		

These platforms support only Layer 2 forwarded packets in the egress VE ACL.

Table 12: Ingress and egress security ACL attachment points

Interface type	ACL type	Ingress support	Egress support
Physical	IPv4	Yes	Yes
	IPv6		
	MAC		
LAG	IPv4		No
	IPv6		
	MAC		
VE	IPv4		Yes
	IPv6		
	MAC		No
Management interface	IPv4 / IPv6		No

Egress MAC ACLs on VE are supported only for switched packets.

Table 13: RACL attachment points

Interface type	ACL type	Ingress support	Egress support
Control plane	IPv4	Yes	n/a
	IPv6		

Extreme 8520 and Extreme 8720 Platforms

The following tables summarize the qualifiers and attachment points supported for various features on the Extreme 8520 and Extreme 8720 platforms.

Table 14: Ingress security ACL

Type	Qualifier	Action	Scale
IPv4	s-ip, d-ip, ip-protocol(tcp/udp/icmp/igmp/esp/customProtocolNumber), TcpFlags, L4SPort, L4DPort, vlan, dscp	Permit, Deny, Count, f orce-tc, force-pcp, force-dscp, force-dp, mirror	768
IPv6	s-ip, d-ip, ip-protocol(tcp/udp/esp/icmpv6/customProtocolNumber), TcpFlags, L4SPort, L4DPort, vlan, dscp		
MAC	s-mac, d-mac, vlan, ethtype, cos		766

Table 15: Egress security ACL

Type	Qualifier	Action	Scale	Note
IPv4	s-ip, d-ip, ip-protocol(tcp/udp/icmp/igmp/esp/customProtocolNumber), TcpFlags, L4SPort, L4DPort, vlan, dscp	Permit, Deny, Count	511	**TCAM Depth:** The TCAM has a depth of 2048 entries. However, the available user ACL scale is less due to reserved system entries. This scale may decrease further with additional feature implementation .
IPv6	s-ip, d-ip, ip-protocol(tcp/udp/esp/icmpv6/customProtocolNumber), TcpFlags, L4SPort,			

Table 15: Egress security ACL (continued)

Type	Qualifier	Action	Scale	Note
	L4DPort, vlan, dscp			
MAC	s-mac, d-mac, vlan, ethtype, cos			

Table 16: RACL

Type	Qualifier	Action	Scale
IPv4	s-ip, d-ip, ip-protocol(tcp/udp/icmp/igmp/esp/customProtocolNumber), TcpFlags, L4SPort, L4DPort, vlan, dscp	Deny, Permit, New CoS, New DSCP, Queue, Next Hop, Count	768
IPv6	s-ip, d-ip, ip-protocol(tcp/udp/esp/customProtocolNumber), TcpFlags, L4SPort, L4DPort, vlan, dscp		

For routed packets, egress ACLs attached on VE work only for VE 1 to 4095. Egress MAC ACLs attached on VE work only for switched packets.

Table 17: Ingress and egress security ACL attachment points

Interface type	ACL type	Ingress support	Egress support
Physical	IPv4	Yes	Yes
	IPv6		
	MAC		
LAG	IPv4		No
	IPv6		
	MAC		

Table 17: Ingress and egress security ACL attachment points (continued)

Interface type	ACL type	Ingress support	Egress support
VE	IPv4		Yes
	IPv6		
	MAC		
Management interface	IPv4 / IPv6		No

Table 18: RACL attachment points

Interface type	ACL type	Ingress support	Egress support
Control plane	IPv4	Yes	n/a
	IPv6		

ACL Platform Limitations

The following sections document known limitations for specific hardware platforms.

Note
Limitations marked with a CSP reference are under active investigation with Broadcom. Check the referenced CSP case for the latest status and any available workarounds.

Jericho J2 DNX Platform

The following limitations apply to ACL features on the Jericho J2 DNX (J2) platform.

- OuterVLAN Priority – Force-PCP Option (OuterVlanPrioNew):

  The OuterVlanPrioNew action with the Force-PCP option is currently not supported on the J2 platform. This action is not functioning as expected. A CSP has been raised with Broadcom to address this issue.

  CSP Reference: CS00012435468

- Force-DSCP Action – Switched Packet Limitation:

  Force-DSCP is supported only in the L3 Ingress Security Group. The ACL QoS Force-DSCP action is implemented using the bcmFieldActionNetworkQos PMF action. Currently, this action applies only to routed packets, as the NetworkQoS field carries the DSCP value only during routing scenarios.

  A CSP has been opened with Broadcom to investigate whether an alternate configuration can enable DSCP modification for switched packets as well.

  CSP Reference: CS00012442197

TD3 Platform

The following limitation applies to ACL features on the TD3 platform.

ACL Remarking – CoS Value Derivation for Routed Traffic:

In ACL remarking scenarios, the CoS value is derived from the traffic class rather than from the ACL Force-PCP value for routed traffic. This appears to be inherent Broadcom platform behavior.

Security ACLs on the Management Interface

Apply a Security ACL to the management interface to restrict management-plane access and optionally limit concurrent client connections.

You use management interface ACLs to restrict inbound access to management services (for example, SSH, gRPC, or SNMP) for IPv4 and IPv6. The switch operating system networking stack enforces management interface ACLs.

Management interface ACLs apply only to management interface ingress. Only the management interface supports connection limiting.

Connection Limiting

Configure a per-rule connection limit to control the number of concurrent connections.

Connection limiting is intended to reduce management-plane resource exhaustion and limit abusive client behavior.

Operational Constraints

- Only one ACL can be attached to the management interface at a time.
- Rules with unsupported match fields or actions are rejected for management interface attachment.
- Rules using the connection-limit field are rejected when applied to non-management interfaces.

Configure a Security ACL on the Management Interface

Create an IPv4 or IPv6 Security ACL and attach it to the management interface ingress to control management-plane access.

You must have administrative privileges.

If you are using external AAA, confirm that access to management services remains available from at least one trusted client subnet.

Management interface ACLs are evaluated on ingress. Only IPv4 and IPv6 match criteria are supported, and only one ACL can be attached at a time.

For supported match fields and actions, see Management Interface ACL Constraints on page 38.

1. Create an IPv4 or IPv6 access list.

   For the exact command syntax, see the ACL CLI reference for your release.
2. Add access control entries (ACEs) to permit or deny traffic to management services.

   Optionally configure a connection limit on applicable ACEs. Connection limiting is supported only for management interface attachments.
3. Attach the ACL to the management interface ingress.

   At any time, only one ACL can be attached to the management interface.
4. Verify ACL statistics.

   Use the ACL statistics show commands to confirm expected matches.

The management interface enforces the configured ACL for new inbound management-plane connections.

If you lose access, use console access or out-of-band procedures appropriate for your deployment.

Management Interface ACL Constraints

Supported ACL types, match fields, actions, and limitations when attaching a Security ACL to the management interface.

Supported ACL types

ACL type	Supported	Notes
IPv4	Yes	Supported for management interface ingress.
IPv6	Yes	Supported for management interface ingress.
MAC	No	Not supported for management interface attachment.

Supported match fields

The following match fields are supported for management interface ACL rules:

- Source IPv4 or IPv6 address
- Destination IPv4 or IPv6 address (where applicable)
- IP protocol
- Layer 4 source port and destination port

Supported actions

- Permit
- Deny
- Count (statistics)

Connection limiting

Connection limiting can be configured on IPv4 and IPv6 ACL rules when the ACL is attached to the management interface.

- Rules with a connection limit are rejected on non-management interface attachment points.
- Connection limiting applies to new connections. Existing connections are not affected.

Attachment limitations

- You can attach them only in the ingress direction.
- You can attach only one ACL to the management interface at a time.
- Unsupported match fields or actions cause validation failure for management interface attachment.

Validation behavior

The system validates management interface ACL rules at configuration time. If a rule includes an unsupported field or action, the system rejects the configuration for the management interface attachment.

Connection Limit on Management Interface

You can configure connection limits on management interfaces for IPv4 and IPv6 ACLs. This means that you can control incoming connections based on IP addresses, protocol, and port numbers.

Key Points

You can:

- Create ACE rules with connection limits for management interface.
- Attach/detach ACLs to management interface ingress.
- Display packet and byte statistics for ACE rules.
- Use for IPv4 and IPv6 addresses, protocol, and port numbers.

Supported Features

- IPv4 and IPv6 ACLs on the management interface
- Connection limit field in Extreme YANG extensions for IPv4 and IPv6 ACLs
- CLI and GNMI

Limitations

- Only supported on the management interface controlled by Linux in Extreme OS ONE.
- Not supported on the VRF on the switch Ethernet port.
- No logging support.

Validation

The system performs the following validations when attaching an ACL rule to an interface:

- Management Interface
  - Verifies rules using only supported fields (IPv4/IPv6 addresses, protocol, ports, and connection limit)
  - Rejects an ACL if any rule contains an unsupported field
- Other Interfaces
  - Verify rules except for the `connlimit` field
  - Reject an ACL if any rule contains the `connlimit` field

CLI Commands

The following CLI commands are used for managing ACLs on the management interface. For details on command syntax, parameters, and usage guidelines, see the *Extreme OS ONE SR Command Reference Guide*.

- Create an ACL

  Command: **device(config)# [no] { ipv4 | ipv6 } access-list <ACL_NAME>**

  Example output:

```
device(config)# ipv4 access-list v4acl
device(config-ip-acl)# exit
device(config)# ipv6 access-list v6acl
device(config-ipv6-acl)#
```

- Add or remove rules (permit/deny, with/without connection limit)

  Command:
  - For IPv4

    [seq <1-65535>] permit (<1-254> | ipv4 | icmp | igmp | esp| udp | tcp [ack] [cwr] [ece] [fin] [push] [reset] [sync] [urg]) (<SRC_IP> <MASK> | <SRC_IP/PREFIX> | any) (<DST_IP> <MASK> | <DST_IP/PREFIX> | any) [dport <1-65535>] [sport <1-65535>] [dscp <0-63>] [vlan <0-4095>] [count] [mirror <NAME>] [force-tc <0-7>] [force-dscp <0-63>] [force-dp <0-2>] [force-pcp <0-7>] [connlimit <1-65535>]

    [seq <1-65535>] deny (<1-254> | ipv4 | icmp | igmp | esp| udp | tcp [ack] [cwr] [ece] [fin] [push] [reset] [sync] [urg]) (<SRC_IP> <MASK> | <SRC_IP/PREFIX> | any) (<DST_IP> <MASK> | <DST_IP/PREFIX> | any) [dport <1-65535>] [sport <1-65535>] [dscp <0-63>] [vlan <0-4095>] [count] [connlimit <1-65535>]
  - For IPv6

    [seq <1-65535>] permit (<1-254> | ipv6 | icmpv6 | esp | udp | tcp [ack] [cwr] [ece] [fin] [push] [reset] [sync] [urg]) (<SRC_IP> <MASK> | <SRC_IP/PREFIX> | any) (<DST_IP> <MASK> | <DST_IP/PREFIX> | any) [dport <1-65535>] [sport <1-65535>] [dscp <0-63>] [vlan

<0-4095>] [count] [mirror <NAME>] [force-tc <0-7>] [force-dscp <0-63>] [force-dp <0-2>] [force-pcp <0-7>] [connlimit <1-65535>]

[seq <1-65535>] deny (<1-254> | ipv6 | icmpv6 | esp | udp | tcp [ack] [cwr] [ece] [fin] [push] [reset] [sync] [urg]) (<SRC_IP> <MASK> | <SRC_IP/PREFIX> | any) (<DST_IP> <MASK> | <DST_IP/PREFIX> | any) [dport <1-65535>] [sport <1-65535>] [dscp <0-63>] [vlan <0-4095>] [count] [connlimit <1-65535>]

Example output:

```
device(config)# ipv6 access-list v6acl
device(config-ipv6-acl)# permit tcp 1::0/64 2::0 ffff::0 sync log
device(config-ipv6-acl)# exit
device(config)# ipv4 access-list v4acl
device(config-ip-acl)# permit ipv4 1.1.1.1/32 2.2.2.0 255.255.255.0 count mirror
test_mirror force-dscp 5 force-dp 1 force-tc 2
device(config-ip-acl)#
device(config)# ipv4 access-list v4limit
device(config-ip-acl)# permit ipv4 10.37.32.18 255.255.255.255 10.37.34.84
255.255.255.255 dport 22 count connlimit 3
device(config-ip-acl)#
```

- Attach or detach ACL to the management interface ingress (applies iptables rules immediately and on reboot)

  Command:

  device(config-if-mgmt-0)# [no] (ipv4 | ipv6 | mac) access-list <ACL_NAME> (in)

  Example output:

```
device(config)# interface management 0
device(config-if-mgmt-0)# ipv4 access-list v4acl in
device(config-if-mgmt-0)# ipv6 access-list v6acl in
device(config-if-mgmt-0)#
```

- Show ACL rule counters

  Command: **show <ip/ipv6/mac> access-list <ACL_NAME/all>**

  Example output:

```
device# show ip access-list all
ip access-list a1
  seq 10 permit ip any any vlan 10 ( 0 Packets, 0 Bytes )
  seq 20 permit tcp any any vlan 100 count ( 0 Packets, 0 Bytes )
ip access-list ipv1
  seq 10 permit ip any any qos-forwarding-group Q1F1 ( 0 Packets, 0 Bytes )
device#
```

- Clear ACL rule counters

  Command: **clear counters <ipv4/ipv6/mac> access-list <ACL_NAME/all>**

  Example output:

```
device# clear counters mac access-list all
device# clear counters ipv6 access-list all
device# clear counters ipv4 access-list a1
device#
```

Supported Rule Fields

ACL rules support a defined set of match fields. The supported fields depend on the interface to which the ACL is applied.

Supported Rule Fields for Management Interface

When an ACL is attached to the management interface, the following fields are supported:

- Source and destination IPv4 or IPv6 address
- Protocol
- Source and destination ports
- `connlimit`
- `count`

Rules are applied in sequence ID order until a match is found.

Rules in an ACL that is attached to the management interface are applied in the same sequence ID order in which they are configured, until a match is found.

In the case of rules with connection limit `n`, the rule is matched only after `n` connections matching the rule are already established. For connection attempts before this (that is, connections 1 to `n-1`), the rule is not matched and processing continues down the list of rules until a match is found or the end of the list is reached.

If no rule matches a packet, the default action is to permit traffic.

To illustrate, to allow four SSH connections from a specific host and to deny TCP connections from any other host, configure the following rules:

```
permit tcp <source specifiers of host1> <destination specifiers of management interface>
dport 22 connlimit 4 count
permit tcp <source specifiers of host1> <destination specifiers of management interface>
count
deny tcp any any
```

Without the second rule, connections before the connection limit is reached (that is, connections 1 to 3) do not match the first rule and instead match the final rule, resulting in the traffic being denied.

Supported Rule Fields for Non-Management Interfaces

Standard ACL match fields are supported. The `connlimit` field is not supported.

When configuring rules that match specific protocols and ports (for example, TCP, ICMP, SSH, or HTTPS), ensure that the rules account for all traffic required to establish those connections. Protocol dependencies can cause connections to fail even when the primary protocol rule appears correct.

For example, a TCP connection over IPv6 relies on ICMPv6 exchanges for neighbor discovery and other control functions. If a restrictive deny rule for ICMPv6 appears

before the TCP rule in the ACL, it may prevent IPv6 TCP connections from being established, even if the TCP rule specifies a higher sequence ID or connection limit."

Note
Use the per-rule packet and byte counters to identify unexpected traffic drops. If a connection is failing, check whether a preceding rule is matching and dropping the control-plane traffic that the connection depends on. To view counters, use: `device# show [(ipv4 | ipv6 | mac)] access-list (all | <ACL_NAME>)`

Rules with unexpectedly high match counts - especially on deny rules - can indicate that dependent protocol traffic is being blocked before the intended rule is reached.

Validation Rules

- ACLs attached to the management interface are validated.
- You cannot use `connlimit` on other interfaces.

ACL QoS and Mirroring

ACL QoS applies QoS settings to traffic using ACL rules on L2 or L3 ingress. ACL QoS can also override port-level QoS configuration for specific traffic flows.

- Use cases:
  - Traffic arriving on untrusted ports
  - Special flow handling, including overriding port-level QoS configuration for specific traffic flows (even on trusted ports)
- Supported options:
  - **`force-tc`**: Set internal traffic class
  - **`force-dscp`**: Mark outgoing DSCP
  - **`force-pcp`**: Set 802.1p value
  - **`force-dp`**: Set drop precedence

Extreme OS ONE 22.2.2.0 supports ACL mirror configuration. ACL mirroring creates a copy of matching packets and sends them to a configured mirror destination.

The following is an example configuration of ACL mirroring and QoS. In this example, m1 is the name of a mirror session that is already configured on the device:

```
device(config)# ipv4 access-list a1
device(config-ip-acl)# seq 20 permit ipv4 192.0.2.0 255.255.255.0 any count force-tc 2
force-pcp 3 force-dscp 41 force-dp 1 mirror m1
```

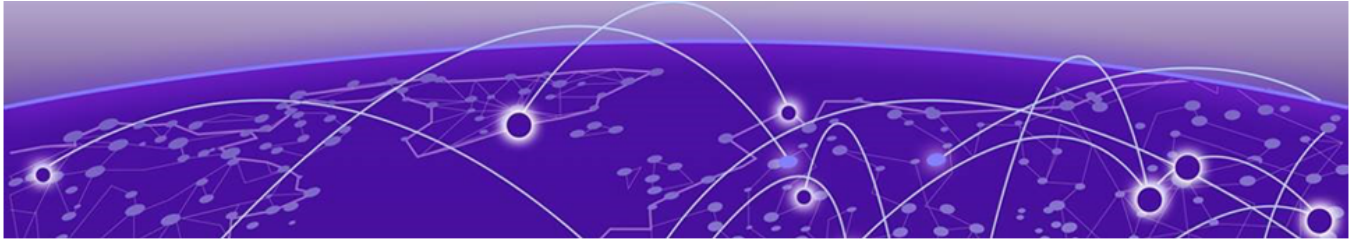
If mirror actions are configured in multiple ACL groups, a separate mirror copy of the packet is generated for each configured mirror destination. Mirror actions are non-conflicting and can operate simultaneously.

For details about mirror sessions, see the "Port Mirroring" chapter in the *Extreme OS ONE SR Management Configuration Guide* as well as the **`mirror`** (global configuration) command in the *Extreme OS ONE SR Command Reference Guide*.

You can see a list of the configured mirror sessions on the device by using the **show running-config mirror** command. In the following example, a single mirror session (in this case, m1) exists on the device:

```
device# show running-config mirror

mirror mirror m1
  description Mirror interface 1 Ethernet 0/1
  destination interface ethernet 0/1
device#
```



User Account and Password Configuration

User Accounts and Roles

A user account defines a user's level of access to the device CLI.

The system uses role-based access control (RBAC) to control user permissions. A role defines which commands a user can execute and with what permissions.. A role is a container for rules, which specify which commands can be executed and with which permissions. When you create a local user account, you must specify a role for that account. The term user refers to any account assigned either the admin or user role.

Extreme OS ONE Switching and Routing supports two roles: admin and user. By default, only the default admin user is provisioned with the admin role. You can configure local users with either admin or user roles.

Default Admin User

The system includes a default administrator account named **admin**. You must change the default password at first login.

Admin accounts can execute all commands supported on the device.

You cannot delete or lock the admin account or configure its password to expire. You can modify its username and password.

Use the following example to change the default admin username and password:

```
device# configure terminal
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# authentication
device(config-system-aaa-authentication)# admin-user mynetworkadmin password !Pass@123!
device(config-system-aaa-authentication)#
```

You can also specify a hashed password by using the **password-hashed** keyword. For details about the **admin-user** command and keywords, see the *Extreme OS ONE SR Command Reference Guide*.

Local Users

User-role accounts can execute all show commands and the following operational commands: **exit**, **ping**, and **traceroute**.

You can delete or lock local user accounts, configure password expiry, and modify usernames and passwords.

The following example creates two local user accounts with the user role:

```
device# configure terminal
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# authentication
device(config-system-aaa-authentication)# user user1 role user password #Pass@456!
inactivity-expiry-period 20 inactivity-warning-period 15
device(config-system-aaa-authentication)# user user2 role user password $Pass@789!
inactivity-expiry-period 20 inactivity-warning-period 15
device(config-system-aaa-authentication)#
```

The following example shows how to create a local user account with the admin role. This example uses the **password-hashed** keyword to specify a hashed password:

```
device# configure terminal
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# authentication
device(config-system-aaa-authentication)# user user3 role
user password-hashed $6$857192462$Cl/kzKrLUO1XTbaAoQ5m62y.5WrrC6AkAxwZ3/
ozLdbkC.DGj.rNMBjwvx7Gqyw0KaP4ciUmuhogS/nKvZDEQ1 inactivity-expiry-period 2 inactivity-
warning-period 1
device(config-system-aaa-authentication)#
```

Force Password Change At First Login

For security, change the default password at first login. No CLI configuration is required. The system automatically prompts the default admin user to update its password upon first successful login after any of the following events:

- ONIE installation
- Factory reset

- Full installation (without preserving settings)
- Copying the default configuration

Note
Forced password change is enabled by default.

Force Password Age Out

To improve security, configure password expiry for all accounts. This section describes how to force users, including admin users, to change their passwords on expiry of a preconfigured time interval. This is a global configuration.

To enforce password expiry:

1. Open a session to access the device.
2. Log in as admin.
3. Access global configuration mode.

```
device# configure terminal
```

4. Configure the maximum password age in days. This time duration is called the *ageout* duration:

```
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# authentication
device(config-system-aaa-authentication)# password-attributes
device(config-system-aaa-authentication-password-attributes)# max-password-age 90
device(config-system-aaa-authentication-password-attributes)#
```

This example sets the maximum password age to 90 days. Each user is forced to change the password every 90 days. This is a global configuration and applies to all local users configured except the default admin user on the system.

Password Expiry Alert

By default, all local users except the admin account receive password expiry alerts. The administrator has the option to customize this feature.

The system generates syslog notifications at configurable severity levels as passwords approach expiration.

Configure password expiry alerts as follows::

```
device# configure terminal
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# authentication
device(config-system-aaa-authentication)# password-attributes
device(config-system-aaa-authentication-password-attributes)# expiry-alert
device(config-system-aaa-authentication-password-attributes-expiry)# info 15
device(config-system-aaa-authentication-password-attributes-expiry)# minor 10
device(config-system-aaa-authentication-password-attributes-expiry)# major 5
device(config-system-aaa-authentication-password-attributes-expiry)# critical 3
device(config-system-aaa-authentication-password-attributes-expiry)#
```

For details about the password expiry alert configuration commands and syntax, see the **password-attributes** command in the *Extreme OS ONE SR Command Reference Guide.*

Set the Password Reuse Policy

You can configure the number of previous passwords that the system checks when validating a new password. This prevents users from reusing previously configured passwords by maintaining a record of the last *n* passwords. For example, if you configure `history 4`, the system prevents reuse of the last four passwords.

If a user tries to reuse a password in violation of this policy, the system issues the following error:

```
%Error: Password was recently used. Please choose a different password
```

This policy is disabled by default. When configured, this policy applies to new passwords and does not affect existing passwords.

To configure the password reuse policy, complete the following steps.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter system configuration mode.

   ```
   device(config)# system
   ```

3. Enter AAA system configuration mode.

   ```
   device(config-system)# aaa
   ```

4. Enter the mode for configuring AAA authentication.

   ```
   device(config-system-aaa)# authentication
   ```

5. Enter the mode for configuring AAA authentication password attributes.

   ```
   device(config-system-aaa-authentication)# password-attributes
   ```

6. Specify the number of prior passwords against which a user's new password is checked.

   ```
   device(config-system-aaa-authentication-password-attributes)# history 5
   ```

   The range is 1 to 10.

7. (Optional) Verify the password history configuration.

   ```
   device(config-system-aaa-authentication-password-attributes)# do show running-config
   system aaa authentication

   system
     aaa
       authentication
         password-attributes
           history 5
         !
       !
     !
   device(config-system-aaa-authentication-password-attributes)#
   ```

The following example shows how to configure the password reuse policy.

```
device# configure terminal
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# authentication
device(config-system-aaa-authentication)# password-attributes
device(config-system-aaa-authentication-password-attributes)# history 5
device(config-system-aaa-authentication-password-attributes)#
```

The following example displays the AAA authentication configuration that is running currently on the device. This example compares users' new password to their 5 prior passwords (including their current password).

```
device# show running-config system aaa authentication

system
  aaa
    authentication
      admin-user admin password-hashed
$6$lOFtgLVtzOEi7jhl$NALdarg9FfowUfWTZUaOnyXG4mehd3cdg3jdGWBFE6uCCSFSHPxstb6Mpcd3UJoA16Upl9
6GZ1pn5ilGURdte.
      user user1 role admin password-hashed $6$518667019$y/
mszgxxTh6NWktOfDi7IkWXTTVuHIBWZxKfcWYZbfJMam8RxF4uUyQCQPMXomVLVK7Ojglfkj5u5S.nyq5ev1
      user user2 role user password-hashed $6$388289609$hhtu7WjjWi8RBaNYM2IcI/WWIo.ct/
Ci56VqA2Yt13DgFotVpoIQxmkX4wWbY2skSLGzMvnBYky0Ywlhs0RuG1
      password-attributes
        min-length 10
        min-lowercase 3
        min-uppercase 2
        min-numeric 2
        min-special 2
        max-password-age 10
        max-retry 4
        history 5
        max-sequence 4
        max-repeat 3
        lockout-duration 5
        expiry-alert
          info 15
          minor 10
          major 5
          critical 3
        !
      !
    !
  !
!
device#
```

Password Requirement Attributes and Default Password Strength

Extreme OS ONE enforces password policies to improve system security. Extreme OS ONE validates the configured values when users create or update passwords.

Password Entropy

Use password attributes to configure password entropy. This lets you specify a maximum number of consecutive characters and a maximum number of repetitive

characters in new passwords. These two functionalities are disabled by default. When configured, these settings apply to new passwords and do not affect existing passwords.

The `max-sequence` parameter limits the number of consecutive sequential characters. The following examples assume a setting of **`max-sequence 2`** (meaning that sequences longer than two characters are *not* allowed).

- Invalid:
  - abc@123 (contains abc and 123)
  - XYZ@789 (contains X→Y→Z and 7→8→9)
  - test@654 (contains 6→5→4 (reverse sequence))
- Valid:
  - abC@12 (a→b is sequential (2 characters are allowed), but b→C breaks the sequence)
  - PaB@27x (contains no 3-character sequence)
  - Xy@19A (contains only 2-character sequences)

The **`max-repeat`** parameter limits the number of consecutive identical (repeating) characters allowed in a new password. The following examples assume a setting of **`max-repeat 2`** (meaning that more than 2 repeating characters are *not* allowed).

- Invalid:
  - aaaB@123 (contains aaa)
  - Passssss@1 (contains ssssss)
  - XXXY@789 (contains XXX)
- Valid:
  - aaB@1234 (has only two repeating a characters)
  - BBc@678 (has only two B characters)
  - XxYy@135 (has no excessive repeats)

Password Character Minimums

By default, passwords must meet the following requirements:

- At least 8 characters
- At least 1 uppercase character
- At least 1 lowercase character
- At least 1 numeric character
- At least 1 special character

Set the Password Entropy Requirements and Character Minimums

You can customize these requirements by using the applicable commands after using the **password-attributes** command to enter AAA authentication password attributes system configuration (config-system-aaa-authentication-password-attributes) mode.

```
device# configure terminal
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# authentication
device(config-system-aaa-authentication)# password-attributes
device(config-system-aaa-authentication-password-attributes)# history 5
device(config-system-aaa-authentication-password-attributes)# max-sequence 4
device(config-system-aaa-authentication-password-attributes)# max-repeat 3
device(config-system-aaa-authentication-password-attributes)# min-length 10
device(config-system-aaa-authentication-password-attributes)# min-uppercase 2
device(config-system-aaa-authentication-password-attributes)# min-lowercase 3
device(config-system-aaa-authentication-password-attributes)# min-special 2
device(config-system-aaa-authentication-password-attributes)# min-numeric 2
device(config-system-aaa-authentication-password-attributes)#
```

These attributes satisfy the { **password** | **password-hashed** } *password* parameter of the **user** command as well as the **username** (GRUB system configuration) command. For details, see the **user** (AAA authentication system configuration) command and the **username** (GRUB system configuration) command respectively in the *Extreme OS ONE SR Command Reference Guide.*

Password Configuration: Special Characters

Passwords can include any characters supported by the Linux system. However, when entering passwords through the CLI terminal, do not use '|' and '?' characters. This restriction does not apply when you configure passwords through gNMI.

Password Maximum Retry and Lockout Duration Attributes

Configure the maximum number of failed login attempts before the system locks the account. The system counts failed login attempts since the last successful login. If the number of attempts exceeds the configured limit, the account is locked. Configure a lockout duration to automatically unlock the account after a specified time.

By default, if an administrator has set a maximum number of login attempts and then a user exceeds that value, the user account remains locked until an administrator unlocks it manually. An administrator can configure a lockout duration (the length of time until the account unlocks automatically, and then the user can try to log in again).

An administrator configures the maximum number of retries by using the **max-retry** command after using the **password-attributes** command to enter AAA authentication password attributes system configuration (config-system-aaa-authentication-password-attributes) mode:

```
device# configure terminal
device(config)# system
device(config-system)# aaa
```

```
device(config-system-aaa)# authentication
device(config-system-aaa-authentication)# password-attributes
device(config-system-aaa-authentication-password-attributes)# max-retry 4
device(config-system-aaa-authentication-password-attributes)#
```

To configure the lockout duration, use the **lockout-duration** command after using the **password-attributes** command to enter AAA authentication password attributes system configuration (config-system-aaa-authentication-password-attributes) mode:

```
device# configure terminal
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# authentication
device(config-system-aaa-authentication)# password-attributes
device(config-system-aaa-authentication-password-attributes)# lockout-duration 5
device(config-system-aaa-authentication-password-attributes)#
```

For details about password configuration commands and syntax, see the *Extreme OS ONE SR Command Reference Guide*.

Display User Authentication Configurations and Password Attribute Settings

An administrator can use the **show running-config system aaa** command to display the current AAA authentication configuration on the device. This includes the authentication configuration for the admin user and each non-admin user as well as the password attribute settings (if set to nondefault values).

Use the following command to display AAA authentication configuration. In this example, two non-admin users are configured (in addition to the default "admin" user). Also, all password attributes are set to nondefault values:

```
device# show running-config system aaa

system
  aaa
    authentication
      admin-user admin password-hashed
$6$lOFtgLVtzOEi7jhl$NALdarg9FfowUfWTZUaOnyXG4mehd3cdg3jdGWBFE6uCCSFSHPxstb6Mpcd3UJoA16Upl9
6GZ1pn5ilGURdte.
      user user1 role admin password-hashed $6$518667019$y/
mszgxxTh6NWktOfDi7IkWXTTVuHIBWZxKfcWYZbfJMam8RxF4uUyQCQPMXomVLVK7Ojglfkj5u5S.nyq5ev1
      user user2 role user password-hashed $6$388289609$hhtu7WjjWi8RBaNYM2IcI/WWIo.ct/
Ci56VqA2Yt13DgFotVpoIQxmkX4wWbY2skSLGzMvnBYky0Ywlhs0RuG1
      password-attributes
        min-length 10
        min-lowercase 3
        min-uppercase 2
        min-numeric 2
        min-special 2
        max-password-age 10
        max-retry 4
        lockout-duration 5
        expiry-alert
          info 15
          minor 10
          major 5
          critical 3
        !
```

```
      !
    !
  !
!
device#
```

Configure an Account to Disable Automatically Upon Inactivity

Configure account inactivity to automatically disable unused accounts.

Sometimes, you must automatically disable an account that is inactive for a set period of time:

- inactivity-expiry-period: Number of days before the account is disabled
- inactivity-warning-period: Number of days before a warning is generated

After the inactivity warning period, a warning syslog is generated.

For details about the **inactivity-expiry-period** and **inactivity-warning-period** parameters, see the **user** command in the *Extreme OS ONE SR Command Reference Guide*..

Note
The default *admin* account cannot be disabled. For the default admin user, inactivity and password expiry are not applicable and cannot be deleted, but their credentials can be modified. For local users, inactivity and password expiry are applicable. These users can have either an admin role or a user role.

1. In privileged EXEC mode, access global configuration mode.
   ```
   device # configure terminal
   ```
2. Access system configuration mode.
   ```
   device(config)# system
   ```
3. Access AAA system configuration mode.
   ```
   device(config-system)# aaa
   ```
4. Access AAA authentication system configuration mode.
   ```
   device(config-system-aaa)# authentication
   ```
5. Enter the **user** command with the **inactivity-expiry-period** parameter along with the number of days of inactivity, after which the account will automatically be disabled.
   ```
   device(config-system-aaa-authentication)# user aming role admin password Testing@123
   inactivity-expiry-period 30 inactivity-warning-period 20
   ```
   The account named aming is now configured to automatically expire after 30 continuous days of inactivity. This is calculated from the day the account was created or from the last login. An expiry RASlog entry is generated when time crosses the account inactivity expiry period.

Configure an Account with an Inactivity Warning

When defining or editing an account that automatically expires, you can specify a duration after which a warning is generated about the inactivity of the account.

By default, users are not warned about the inactivity of their accounts. Use the **inactivity-warning-period** parameter to configure the number of days after which a warning is generated about the account being inactive. For example, when set to 20 days, a warning is generated when a specific user account is inactive for 20 days.

Without configuring an expiry period, a warning period cannot be configured.

1. In the privileged EXEC mode, access global configuration mode.

```
device # configure terminal
```

2. Enter the **user** command with the **inactivity-warning-period** keyword and the number of days.

```
device(config-system-aaa-authentication)# user aming role user password Testing@123
inactivity-expiry-period 20 inactivity-warning-period 10
```

The account *aming* is now configured to generate a warning after 10 continuous days of the account being inactive. A warning RASLog is generated when the account inactivity warning period expires.

Change Default Password for the System Default Accounts

The default system username is **admin**. Upon first login, the system prompts you to change the default username and password.

The credentials must follow these password requirements:

- Default minimum length: Eight characters
- Complexity:
  - Must include uppercase, lowercase, numeric, and special characters
  - Must include both uppercase and lowercase letters

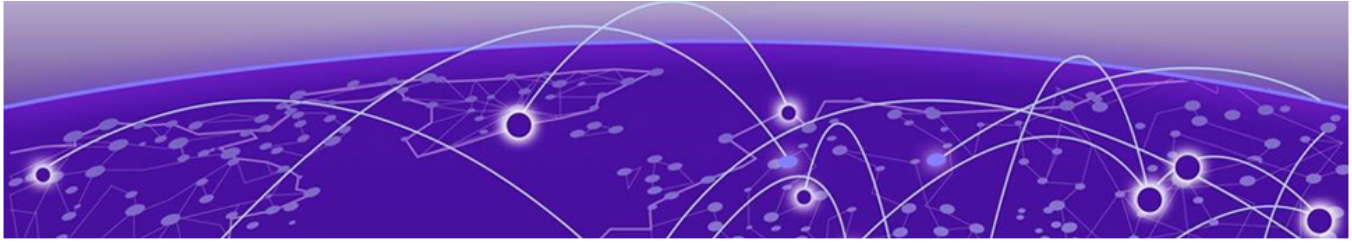
Use the following commands to change the default username or password.

1. To change the password for the 'admin' user, enter the following command:

```
device(config-system-aaa-authentication)# admin-user admin password agt2bna!cdx7N@M.hfp
```

2. To change the default username from 'admin' to a customized username, enter the following command:

```
device(config-system-aaa-authentication)# admin-user test password agt2bna!cdx7N@M.hfp
```



Northbound Interfaces and Security

This section describes northbound interface components.

Northbound Interfaces

SSH

Secure Shell (SSH) provides secure access to management functions on remote devices. Unlike Telnet, SSH provides an encrypted connection. The following lists contain the recommended configurations for SSH.

As a best practice, use the following set of ciphers. Ciphers protect the data transported over an SSH connection. Each cipher is an algorithm that encrypts the link. Each name indicates the algorithm and cryptographic parameters that are used:

aes256-gcm-openssh
aes256-ctr
chacha20-poly1305-openssh
aes192-ctr
aes128-gcm-openssh
aes128-ctr

You use the **cipher** command to configure the SSH server to use a set of ciphers in an order of priority that you specify.

As a best practice, use the following set of message authentication code (MAC) algorithms. Message authentication codes maintain the integrity of each message sent across an SSH connection. Each name indicates the algorithm and cryptographic parameters that are used:

hmac-sha2-512-etm-openssh
hmac-sha2-256-etm-openssh

You use the **mac** command to configure the SSH server to use a set of MAC algorithms specified in a priority order that you specify.

As a best practice, use the following set of key exchange algorithms. Key exchange algorithms are used to securely exchange a shared session key with a peer. Each algorithm distributes a shared key to prevent external interference, manipulation, or recovery. Each name indicates the algorithm and cryptographic parameters that are used:

curve25519-sha256
curve25519-sha256-libssh
diffie-hellman-group-exchange-sha256
diffie-hellman-group16-sha512
diffie-hellman-group18-sha512

You use the **`key-exchange`** command to configure SSH to use a set of key exchange algorithms in a priority order that you specify.

For more information about SSH server and client configuration and other details, see the "Configuration Fundamentals" chapter in the *Extreme OS ONE SR Management Configuration Guide*. For details about the commands in this section, see the *Extreme OS ONE SR Command Reference Guide*.

Other Northbound Interfaces

Any configuration changes apply only to new connections.

- Telnet:
  - Not enabled by default
  - Uses unencrypted communication (port is 23)
  - Shares a combined session limit of 32 with SSH
  - Terminates sessions when a user role is updated, a local user is deleted, or the terminal times out
- GRPC server:
  - Not configured by default
  - Uses TLS for secure communication
  - Requires a certificate ID for configuration
  - Does not enable mutual TLS by default
- Network Time Protocol (NTP):
  - Not configured by default
  - Supports up to 8 NTP servers and 8 peers
- TLS: Supports versions 1.2 and 1.3
- System logging (SYSLOG)
-
  - Not configured by default
  - Uses TLS for secure communication
  - Requires a certificate ID for configuration
  - Does not enable mutual TLS by default

  Extreme OS ONE supports multiple SSH and Telnet instances with unique VRFs.

Password Recovery

- Local user password recovery:

  Forgotten passwords for local users can be reset by logging in as admin user and updating the password.
- Admin user password recovery:

  If the admin user password is forgotten, perform a net install to recover access.

Note
GRUB boot loader password recovery: If the GRUB boot loader password is forgotten, it cannot be recovered through software or CLI means. There is no mechanism to reset or bypass GRUB credentials without physical access to the device and a net install — and depending on the platform configuration, recovery may not be possible at all. In this case, the device will likely require an RMA (Return Merchandise Authorization). To avoid this situation, ensure GRUB credentials are stored securely and documented in your organization's credential management system.

AAA (Authentication, Authorization, and Accounting)

This section describes authentication, authorization, and accounting.

Authentication

This section describes external authentication, common features, and authentication order.

External Authentication

1. TACACS+: TACACS+ is an authentication protocol used to authenticate users through remote servers.
   - Default Settings
     - Port: 49
     - Retry attempts: 2
     - Timeout: 3 seconds
     - Maximum servers: 6
     - Default role: "user"
   - Configuration Requirements
     - The shared key must match the TACACS+ server configuration.
     - Assign a role (admin or user) to each user on the server. This role is fetched from the extreme-role attribute associated with the user on the TACACS+ server. For example:

```
user = npbuser1 {
  default service = permit
  chap = cleartext "password123"
  service = exec {
        priv-lvl=15
 extreme-role = admin
```

```
        }
}
```

2. LDAP: LDAP is an open protocol used for directory services authentication.
   - Default Settings
     - Port: 389 (LDAP), 636 (secure LDAP)
     - Retry attempts: 2
     - Timeout: 3 seconds
     - Maximum servers: 6
     - Default role: "user"
   - Configuration Requirements
     - The LDAP server's CA certificates must be imported into Extreme OS ONE.
     - Role mapping should be performed to map LDAP user roles to available roles in Gen4OS. The **`map-role`** command maps the group of the user at the LDAP server. For details, see the *Extreme OS ONE SR Command Reference Guide*.
3. RADIUS: RADIUS is a networking protocol that authenticates remote management users.
   - Default Settings
     - Port: 1812 (RADIUS over UDP), 2083 (RADIUS over TLS)
     - Retry attempts: 2
     - Timeout: 3 seconds
     - Maximum servers: 6
     - Default role: "user"
   - Configuration Requirements
     - The RADIUS server's CA certificate must be imported into Extreme OS ONE.
     - Assign a role (admin or user) to each user on the server. ATTRIBUTE under VENDOR on the RADIUS server determines the role. See the following examples for the location:
       - /usr/share/freeradius/dictionary.extreme file:

```
VENDOR          Extreme                         1916
ATTRIBUTE       Extreme-Security-Profile        212     string
```

       - /usr/share/freeradius/dictionary.brocade file:

```
VENDOR          Brocade                         1588
ATTRIBUTE       Brocade-Auth-Role               1       string
```

Common Features

- In-band Support: Configure each external administration server with a VRF and source interface.
- Failover: The system fails over to another server if a server does not respond or becomes unreachable.

Authentication Order

Authentication mode defines the order of the authentication sources.

- Default Mode: Local authentication only.
- Configurable Modes: TACACS+, LDAP, or RADIUS with local fallback.
- Applies to: SSH, Telnet, and gNMI.

Authorization

- Role-Based Access Control (RBAC): Controls access based on assigned roles.
  - Admin (read-write access)
  - User (read-only access)
  - Role assignment: Specify a role when creating a user account

Accounting

- Local Accounting: Logs device operations locally (view using `show logging`).
- Remote Accounting: Supported for TACACS+ and RADIUS.
- Logs device operations locally (view using `show logging`).
- Attributes:
  - `cmd`: Command string
  - `status`: Execution status
- gNMI limitation: gNMI activity is not logged to RADIUS.

CLIs for AAA Configuration

The following sections describe how to configure the AAA settings on a device. They include settings for the accounting method (TACACS+ or RADIUS), event logging, authentication method (such as LDAP or RADIUS), credentials and roles, password requirements, and server configuration.

Configure AAA Accounting

To configure AAA accounting. This section applies to RADIUS and TACACS+ AAA.

For details about the commands in this section, see the *Extreme OS ONE SR Command Reference Guide.*

1. Enter global configuration mode:

```
device# configure terminal
```

2. Enter system configuration mode:

```
device(config)# system
```

3. Enter AAA configuration mode:

```
device(config-system)# aaa
```

4. Enter AAA accounting configuration mode:

```
device(config-system-aaa)# accounting
```

5. Specify the accounting method. You can specify either radius or tacacs+.

```
device(config-system-aaa-accounting)# accounting-method radius
```

6. Specify the event types that need to be part of the accounting information. You can specify login types, command types, or both.

```
device(config-system-aaa-accounting)# event login
device(config-system-aaa-accounting)# event command
```

7. (Optional) Confirm the configuration:

```
device# show running-config system aaa

system
  aaa
    accounting
      accounting-method tacacs+
      event login
      event command
    authentication
      admin-user admin password-hashed $6$/FuU/
eF7R6Mzy.0m$7EYOmd9u1b.IkA9cwquFigmCmHHAzHwJ0j8o0goPeDSg0twsdYDb8bAFJMsg6vkvxn/
4XfND4urMl/EAQB8a1/
      authentication-method tacacs+ local
      password-attributes
        expiry-alert
          info 15
          minor 10
          major 5
          critical 3
        history 5
        max-sequence 4
        max-repeat 3
        lockout-duration 5
      user
        user admin1 role admin password-hashed
$6$9WWilKua$GRF54FTAYadiW7ZsXMqc6lp6dIyOEPgQO5cr6f1kotX/
XKGfCPjX.DwodXNSq9HM7GXSzTWKvDgrzBUF43SQV/ inactivity-expiry-period 20 inactivity-
warning-period 15
        user user1 role user password-hashed
$6$aEGceoh1$UmCyPNR5Rfejo5MiEadS8CiZv.NCXFG8MCY4nn914VYC/6OdA9fVmDbyuQWevyONLHdqLbVeIb/
9xi9xWBACi. inactivity-expiry-period 15 inactivity-warning-period 10
      !
    server-group tacacs+
      server 192.0.2.0
       secret-key-hashed
vZhG7vy09QiPuOz1qBY7hAKEU6p28k2Y9OYbXuKKSaXBBLGn5cGS526d55tecsopUWMlOEWu4xPLo2JOwerGEg=
=
        source-interface ethernet 0/1
        vrf tenant1
    token-validator val1
      ssl-profile-id sp1
    !
  !
!
```

Configure AAA Authentication

To configure AAA authentication.

1. Enter global configuration mode:

```
device# configure terminal
```

2. Enter system configuration mode:

```
device(config)# system
```

3. Enter AAA system configuration mode:

```
device(config-system)# aaa
```

4. Enter AAA authentication system configuration mode:

```
device(config-system-aaa)# authentication
```

5. Specify a nondefault password for the admin user. You can optionally specify a nondefault name for the admin user (the default name is "admin") if needed.

```
device(config-system-aaa-authentication)# admin-user admin password Admin45#1!
```

6. Specify the authentication method. You can specify **ldap**, **radius**, or **tacacs+**. You must include the **local** keyword.

```
device(config-system-aaa-authentication)# authentication-method ldap local
```

7. (Optional) Create a user with admin privileges. The inactivity periods (in days) are optional.

```
device(config-system-aaa-authentication)# user admin1 role admin password Admin46#1!
inactivity-expiry-period 20 inactivity-warning-period 15
```

8. (Optional) Create a user with user privileges. The inactivity periods (in days) are optional.

```
device(config-system-aaa-authentication)# user user1 role user password User45#1!
inactivity-expiry-period 15 inactivity-warning-period 10
```

9. (Optional) Confirm the configuration:

```
device# show running-config system aaa

system
  aaa
    accounting
      accounting-method tacacs+
      event login
      event command
    authentication
      admin-user admin password-hashed $6$/FuU/
eF7R6Mzy.0m$7EYOmd9u1b.IkA9cwquFigmCmHHAzHwJ0j8o0goPeDSg0twsdYDb8bAFJMsg6vkvxn/
4XfND4urMl/EAQB8a1/
      authentication-method tacacs+ local
      password-attributes
        expiry-alert
          info 15
          minor 10
          major 5
          critical 3
        history 5
        max-sequence 4
        max-repeat 3
        lockout-duration 5
      user
        user admin1 role admin password-hashed
$6$9WWilKua$GRF54FTAYadiW7ZsXMqc6lp6dIyOEPgQO5cr6f1kotX/
XKGfCPjX.DwodXNSq9HM7GXSzTWKvDgrzBUF43SQV/ inactivity-expiry-period 20 inactivity-
warning-period 15
        user user1 role user password-hashed
```

```
$6$aEGceoh1$UmCyPNR5Rfejo5MiEadS8CiZv.NCXFG8MCY4nn914VYC/6OdA9fVmDbyuQWevyONLHdqLbVeIb/
9xi9xWBACi. inactivity-expiry-period 15 inactivity-warning-period 10
      !
    server-group tacacs+
      server 192.0.2.0
        secret-key-hashed
vZhG7vy09QiPuOz1qBY7hAKEU6p28k2Y9OYbXuKKSaXBBLGn5cGS526d55tecsopUWMlOEWu4xPLo2JOwerGEg=
=
        source-interface ethernet 0/1
        vrf tenant1
    token-validator val1
      ssl-profile-id sp1
    !
  !
!
```

Configure AAA Password Attributes

This section describes how to configure AAA password attributes. These settings apply only to new passwords.

1. Enter global configuration mode:

   ```
   device# configure terminal
   ```

2. Enter system configuration mode:

   ```
   device(config)# system
   ```

3. Enter AAA system configuration mode:

   ```
   device(config-system)# aaa
   ```

4. Enter AAA authentication system configuration mode:

   ```
   device(config-system-aaa)# authentication
   ```

5. Enter AAA authentication password attributes system configuration mode:

   ```
   device(config-system-aaa-authentication)# password-attributes
   ```

6. Enter AAA authentication password attributes expiry alert system configuration mode:

   ```
   device(config-system-aaa-authentication-password-attributes)# expiry-alert
   ```

7. Specify the info, minor, major, and critical expiry alert periods (in days):

   ```
   device(config-system-aaa-authentication-password-attributes-expiry)# info 15
   device(config-system-aaa-authentication-password-attributes-expiry)# minor 10
   device(config-system-aaa-authentication-password-attributes-expiry)# major 5
   device(config-system-aaa-authentication-password-attributes-expiry)# critical 3
   ```

8. Specify the requirements for password strings. The default is one character of each type.

   ```
   device(config-system-aaa-authentication-password-attributes-expiry)# exit
   device(config-system-aaa-authentication-password-attributes)# min-length 10
   device(config-system-aaa-authentication-password-attributes)# min-lowercase 3
   device(config-system-aaa-authentication-password-attributes)# min-uppercase 2
   device(config-system-aaa-authentication-password-attributes)# min-numeric 2
   device(config-system-aaa-authentication-password-attributes)# min-special 2
   ```

9. Specify the maximum number of days that can elapse before a password must be changed. The default is zero.

```
device(config-system-aaa-authentication-password-attributes)# max-password-age 10
```

10. Specify the allowed number of password retries before an account is locked. The default is zero retries.

```
device(config-system-aaa-authentication-password-attributes)# max-retry 4
```

11. Specify the lockout duration (in minutes). The default is zero minutes.

```
device(config-system-aaa-authentication-password-attributes)# lockout-duration 5
```

12. (Optional) Confirm the configuration:

```
device# show running-config system aaa

system
  aaa
    accounting
      accounting-method tacacs+
      event login
      event command
    authentication
      admin-user admin password-hashed $6$/FuU/
eF7R6Mzy.0m$7EYOmd9u1b.IkA9cwquFigmCmHHAzHwJ0j8o0goPeDSg0twsdYDb8bAFJMsg6vkvxn/
4XfND4urMl/EAQB8a1/
      authentication-method tacacs+ local
      password-attributes
        expiry-alert
          info 15
          minor 10
          major 5
          critical 3
        min-length 10
        min-lowercase 3
        min-uppercase 2
        min-numeric 2
        min-special 2
        max-password-age 10
        max-retry 4
        lockout-duration 5
      user
        user admin1 role admin password-hashed
$6$9WWilKua$GRF54FTAYadiW7ZsXMqc6lp6dIyOEPgQO5cr6f1kotX/
XKGfCPjX.DwodXNSq9HM7GXSzTWKvDgrzBUF43SQV/ inactivity-expiry-period 20 inactivity-
warning-period 15
        user user1 role user password-hashed
$6$aEGceoh1$UmCyPNR5Rfejo5MiEadS8CiZv.NCXFG8MCY4nn914VYC/6OdA9fVmDbyuQWevyONLHdqLbVeIb/
9xi9xWBACi. inactivity-expiry-period 15 inactivity-warning-period 10
      !
    server-group tacacs+
      server 192.0.2.0
        secret-key-hashed
vZhG7vy09QiPuOz1qBY7hAKEU6p28k2Y9OYbXuKKSaXBBLGn5cGS526d55tecsopUWMlOEWu4xPLo2JOwerGEg=
=
        source-interface ethernet 0/1
        vrf tenant1
    token-validator val1
      ssl-profile-id sp1
    !
  !
!
```

Configure LDAP AAA Server Group

To configure a AAA server group for LDAP.

1. Enter global configuration mode:

```
device# configure terminal
```

2. Enter system configuration mode:

```
device(config)# system
```

3. Enter AAA configuration mode:

```
device(config-system)# aaa
```

4. Specify an LDAP server group and enter AAA LDAP server group configuration mode.

```
device(config-system-aaa)# server-group ldap
```

5. Configure a role mapping to a group that is configured on the LDAP server. You can specify either **admin** or **user** as the role.

```
device(config-server-group-ldap)# map-role group group1 role admin
```

6. Specify the LDAP server IP address:

```
device(config-server-group-ldap)# server 192.0.2.0
```

7. Specify a shared secret for authentication on the LDAP server host.

```
device(config-server-group-ldap-192.0.2.0)# secret-key sharedsecret1
```

8. Specify the source of LDAP protocol packets:

```
device(config-server-group-ldap-192.0.2.0)# source-interface ethernet 0/1
```

9. Specify the VRF instance:

```
device(config-server-group-ldap-192.0.2.0)# vrf tenant1
```

10. Specify a base domain name. This lets you use the base domain name to perform search operations in the active directory tree.

```
device(config-server-group-ldap-192.0.2.0)# base-dn example.com
```

11. Enable LDAP over TLS:

```
device(config-server-group-ldap-192.0.2.0)# ldaps
```

12. Specify the ID of your SSL profile that is associated with the LDAP server:

```
device(config-server-group-ldap-192.0.2.0)# ssl-profile-id ldap-sp1
```

13. (Optional) Confirm the configuration:

```
device# show running-config system aaa

system
  aaa
    accounting
      event login
      event command
    authentication
      admin-user admin password-hashed $6$/FuU/
eF7R6Mzy.0m$7EYOmd9u1b.IkA9cwquFigmCmHHAzHwJ0j8o0goPeDSg0twsdYDb8bAFJMsg6vkvxn/
4XfND4urMl/EAQB8a1/
      authentication-method ldap local
      password-attributes
        expiry-alert
          info 15
```

```
          minor 10
          major 5
          critical 3
        history 5
        max-sequence 4
        max-repeat 3
        lockout-duration 5
      user
        user admin1 role admin password-hashed
$6$9WWilKua$GRF54FTAYadiW7ZsXMqc6lp6dIyOEPgQO5cr6f1kotX/
XKGfCPjX.DwodXNSq9HM7GXSzTWKvDgrzBUF43SQV/ inactivity-expiry-period 20 inactivity-
warning-period 15
        user user1 role user password-hashed
$6$aEGceoh1$UmCyPNR5Rfejo5MiEadS8CiZv.NCXFG8MCY4nn914VYC/6OdA9fVmDbyuQWevyONLHdqLbVeIb/
9xi9xWBACi. inactivity-expiry-period 15 inactivity-warning-period 10
      !
    server-group ldap
      map-role group group1 role admin
      server 192.0.2.0
        secret-key-hashed
vZhG7vy09QiPuOz1qBY7hAKEU6p28k2Y9OYbXuKKSaXBBLGn5cGS526d55tecsopUWMlOEWu4xPLo2JOwerGEg=
=
        source-interface ethernet 0/1
        vrf tenant1
        base-dn example.com
        ldaps
        ssl-profile-id ldap-sp1
    token-validator val1
      ssl-profile-id sp1
    !
  !
!
```

Configure RADIUS AAA Server Group

To configure a AAA server group for RADIUS.

1. Enter global configuration mode:
   ```
   device# configure terminal
   ```
2. Enter system configuration mode:
   ```
   device(config)# system
   ```
3. Enter AAA configuration mode:
   ```
   device(config-system)# aaa
   ```
4. Specify a RADIUS server group and enter AAA RADIUS server group configuration mode.
   ```
   device(config-system-aaa)# server-group radius
   ```
5. Specify the IP address for the RADIUS server and enter AAA RADIUS server group server configuration mode.
   ```
   device(config-server-group-radius)# server 192.0.2.0
   ```
6. Specify a shared secret for authentication on the RADIUS server host.
   ```
   device(config-server-group-radius-192.0.2.0)# secret-key sharedsecret1
   ```
7. Specify the source of RADIUS protocol packets:
   ```
   device(config-server-group-radius-192.0.2.0)# source-interface ethernet 0/1
   ```

8. Specify the VRF instance:

```
device(config-server-group-radius-192.0.2.0)# vrf tenant1
```

9. Enable RADIUS Security (RadSec) on the RADIUS server host:

```
device(config-server-group-radius-192.0.2.0)# radsec
```

10. (Optional) Confirm the configuration:

```
device# show running-config system aaa

system
  aaa
    accounting
      accounting-method radius
      event login
      event command
    authentication
      admin-user admin password-hashed $6$/FuU/
eF7R6Mzy.0m$7EYOmd9u1b.IkA9cwquFigmCmHHAzHwJ0j8o0goPeDSg0twsdYDb8bAFJMsg6vkvxn/
4XfND4urMl/EAQB8a1/
      authentication-method radius local
      password-attributes
        expiry-alert
          info 15
          minor 10
          major 5
          critical 3
        history 5
        max-sequence 4
        max-repeat 3
        lockout-duration 5
      user
        user admin1 role admin password-hashed
$6$9WWilKua$GRF54FTAYadiW7ZsXMqc6lp6dIyOEPgQO5cr6f1kotX/
XKGfCPjX.DwodXNSq9HM7GXSzTWKvDgrzBUF43SQV/ inactivity-expiry-period 20 inactivity-
warning-period 15
        user user1 role user password-hashed
$6$aEGceoh1$UmCyPNR5Rfejo5MiEadS8CiZv.NCXFG8MCY4nn914VYC/6OdA9fVmDbyuQWevyONLHdqLbVeIb/
9xi9xWBACi. inactivity-expiry-period 15 inactivity-warning-period 10
      !
    server-group radius
      server 192.0.2.0
        secret-key-hashed
vZhG7vy09QiPuOz1qBY7hAKEU6p28k2Y9OYbXuKKSaXBBLGn5cGS526d55tecsopUWMlOEWu4xPLo2JOwerGEg=
=
        source-interface ethernet 0/1
        vrf tenant1
        radsec
    token-validator val1
      ssl-profile-id sp1
    !
  !
!
```

Configure TACACS+ AAA Server Group

To configure group for TACACS+.

1. Enter global configuration mode:

```
device# configure terminal
```

2. Enter system configuration mode:

```
device(config)# system
```

3. Enter AAA configuration mode:

```
device(config-system)# aaa
```

4. Specify a TACACS+ server group and enter AAA TACACS+ server group configuration mode:

```
device(config-system-aaa)# server-group tacacs+
```

5. Configure a role mapping to a group that is configured on the TACACS+ server. You can specify either **admin** or **user** as the role.

```
device(config-server-group-tacacs+)# map-role group group1 role admin
```

6. Specify the IP address for the TACACS+ server and enter AAA TACACS+ server group server configuration mode:

```
device(config-server-group-tacacs+)# server 192.0.2.0
```

7. Specify a shared secret for authentication on the TACACS+ server host:

```
device(config-server-group-tacacs+-192.0.2.0)# secret-key sharedsecret1
```

8. Specify the source of TACACS+ protocol packets:

```
device(config-server-group-tacacs+-192.0.2.0)# source-interface ethernet 0/1
```

9. Specify the VRF network instance within which the TACACS+ server is listening:

```
device(config-server-group-tacacs+-192.0.2.0)# vrf tenant1
```

10. (Optional) Confirm the configuration:

```
device# show running-config system aaa

system
  aaa
    accounting
      accounting-method tacacs+
      event login
      event command
    authentication
      admin-user admin password-hashed $6$/FuU/
eF7R6Mzy.0m$7EYOmd9u1b.IkA9cwquFigmCmHHAzHwJ0j8o0goPeDSg0twsdYDb8bAFJMsg6vkvxn/
4XfND4urMl/EAQB8a1/
      authentication-method tacacs+ local
      password-attributes
        expiry-alert
          info 15
          minor 10
          major 5
          critical 3
        history 5
        max-sequence 4
        max-repeat 3
        lockout-duration 5
      user
        user admin1 role admin password-hashed
$6$9WWilKua$GRF54FTAYadiW7ZsXMqc6lp6dIyOEPgQO5cr6f1kotX/
XKGfCPjX.DwodXNSq9HM7GXSzTWKvDgrzBUF43SQV/ inactivity-expiry-period 20 inactivity-
warning-period 15
        user user1 role user password-hashed
$6$aEGceoh1$UmCyPNR5Rfejo5MiEadS8CiZv.NCXFG8MCY4nn914VYC/6OdA9fVmDbyuQWevyONLHdqLbVeIb/
9xi9xWBACi. inactivity-expiry-period 15 inactivity-warning-period 10
      !
    server-group tacacs+
```

```
      server 192.0.2.0
        secret-key-hashed
vZhG7vy09QiPuOz1qBY7hAKEU6p28k2Y9OYbXuKKSaXBBLGn5cGS526d55tecsopUWMlOEWu4xPLo2JOwerGEg=
=
        source-interface ethernet 0/1
        vrf tenant1
    token-validator val1
      ssl-profile-id sp1
    !
  !
!
```

Configure AAA Token Validator

Create a token validator instance to validate JSON web tokens (JWTs) in gNMI or gNOI requests.

1. Import the SSL profile.

```
device# certificate-manager import ssl-profile-id sp1 ca-certificate protocol scp host
192.0.2.0 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
```

2. Enter global configuration mode:

```
device# configure terminal
```

3. Enter system configuration mode:

```
device(config)# system
```

4. Enter AAA system configuration mode:

```
device(config-system)# aaa
```

5. Create a token validator and enter token validator aaa system configuration mode:

```
device(config-system-aaa)# token-validator val1
```

6. Specify an SSL profile:

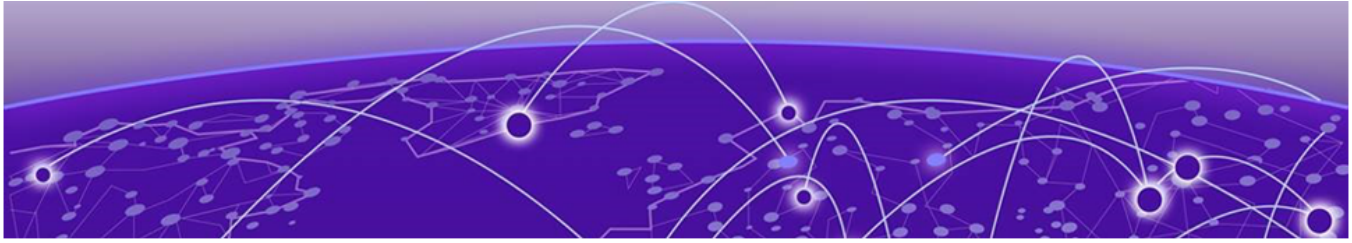
```
device(config-system-aaa-token-validator-val1)# ssl-profile-id sp1
```

7. (Optional) Confirm the configuration:

```
device# show running-config system aaa

system
  aaa
    accounting
      accounting-method tacacs+
      event login
      event command
    authentication
      admin-user admin password-hashed $6$/FuU/
eF7R6Mzy.0m$7EYOmd9u1b.IkA9cwquFigmCmHHAzHwJ0j8o0goPeDSg0twsdYDb8bAFJMsg6vkvxn/
4XfND4urMl/EAQB8a1/
      authentication-method tacacs+ local
      password-attributes
        expiry-alert
          info 15
          minor 10
          major 5
          critical 3
        history 5
        max-sequence 4
        max-repeat 3
        lockout-duration 5
```

```
      user
        user admin1 role admin password-hashed
$6$9WWilKua$GRF54FTAYadiW7ZsXMqc6lp6dIyOEPgQO5cr6f1kotX/
XKGfCPjX.DwodXNSq9HM7GXSzTWKvDgrzBUF43SQV/ inactivity-expiry-period 20 inactivity-
warning-period 15
        user user1 role user password-hashed
$6$aEGceoh1$UmCyPNR5Rfejo5MiEadS8CiZv.NCXFG8MCY4nn914VYC/6OdA9fVmDbyuQWevyONLHdqLbVeIb/
9xi9xWBACi. inactivity-expiry-period 15 inactivity-warning-period 10
      !
    server-group tacacs+
      server 192.0.2.0
        secret-key-hashed
vZhG7vy09QiPuOz1qBY7hAKEU6p28k2Y9OYbXuKKSaXBBLGn5cGS526d55tecsopUWMlOEWu4xPLo2JOwerGEg=
=
        source-interface ethernet 0/1
        vrf tenant1
    token-validator val1
      ssl-profile-id sp1
    !
  !
!
```



Policy-Based Routing

Use policy-based routing to overrule regular destination-based routing by creating custom rules (policies) to direct traffic based on indicators such as the source IP address, destination IP address, port, or protocol, rather than a destination address. These policies provide more control for QoS, security, and load balancing to allow specific traffic (such as VoIP) to use a low-latency link while limiting other traffic (such as file transfers) to a high-bandwidth link to prioritize mission-critical applications or restrict them to specific paths.

Routing Policy Overview

Routing policies control route placement in and advertisement from routing tables. Two major components are involved:

- Routing Policy Server
  - Exists within the Classifier Microservice
  - Handles configuration commands for route-filtering objects
  - Validates and processes commands, updating State DB
- Routing Policy Library
  - Provides APIs for client microservices (routing protocols) to apply routing filters
  - Maintains a per-client database of route-filtering object information
  - Supports diverse routing filters and route filtering logic

Policy Control Points

Routing policies can control routing information at two points: before placement in the routing table or after placement in the routing table.

Client Microservice Interaction

The client microservice handles configuration commands for applying the routing policy on the desired control point of a protocol. Client microservices register with the routing policy library and use its APIs to complete the following tasks:

- Apply routing filters.

- Evaluate routes against routing policies.
- Augment or change advertised or accepted route information.

CLIs for Routing Policy Configuration

To configure routing policies that filter routes and manipulate BGP attributes. For details on command syntax, parameters, and usage guidelines, see the *Extreme OS ONE SR Command Reference Guide*.

Create Routing Policy Building Blocks

Routing policy statements reference objects such as prefix sets, AS-path sets, and community sets to perform matching and apply actions. Before you can build a policy, you must create these foundational components.

Use the following procedure to create prefix sets and BGP-defined sets that serve as reusable components for routing policies.

1. **configure terminal**

   Enter global configuration mode to begin defining routing policy objects.
2. **route-policy**

   Enter routing policy configuration mode. Nothing configured here is active yet; this step prepares reusable data structures.
3. **prefix-set** *prefix-name*

   Create a prefix set to group destination networks for use in routing policy match conditions.

```
prefix-set prefix1
 prefix 192.0.2.0/24 mask-range exact
 prefix 192.0.3.0/24 mask-range 25..30
 prefix 2001:db8:1::/64
 exit
```

4. **bgp-defined-sets**

   Enter the BGP-defined sets mode. You use these sets to create BGP policies to match on AS paths, communities, and extended communities.
5. **as-path-set** *name*

   Define an AS-path set. These sets let policies match routes based on where they have traveled in the AS-path.

```
as-path-set aspath1
 member 65535 65400
 exit
```

6. **community-set** *name*

   Create a community set for matching routes tagged with specific BGP community values.

```
community-set comm1
 match-set-options all
 member 65535:100
 exit
```

7. **ext-community-set** *name*

   Create an extended community set to make VPN route targets available to routing policies.

```
ext-community-set extcomm1
 match-set-options all
 member route-target:65000:1
 exit
```

You can now reference these routing policy building blocks in routing policy statements. (No routing impact occurs until you attach them to routing processes such as BGP.)

Create a Routing Policy and Statements

A routing policy consists of one or more ordered statements. Each statement matches specific route attributes and applies actions to matched routes.

Use the following procedure to define a routing policy, configure match conditions, and apply BGP-specific actions.

1. **configure terminal**

   Enter global configuration mode.
2. **route-policy**

   Enter routing policy configuration mode.
3. **policy** *policy-name*

   Create a routing policy container. A policy can include multiple statements that run in ascending sequence order.
4. **statement** *sequence*

   Create a policy statement. Lower sequence numbers are evaluated first.
5. **conditions**

   Enter the match conditions submode for this statement. Only routes matching these conditions will move on to the actions stage.

6. **bgp-conditions**

   Enter the BGP conditions mode to match routes based on BGP attributes.

   ```
   match-as-path-set aspath1 any
   match-community-set comm1
   exit
   ```

7. **match-prefix-set** *prefix-set* **any**

   Add a prefix-based match condition. Matching both prefix and BGP attributes enables very targeted routing behavior.

8. **actions**

   Enter the actions mode. Actions modify the attributes of matching routes.

9. **bgp-actions**

   Enter the BGP actions mode to apply BGP-specific route modifications.

   ```
   set-as-path-prepend 110 5
   set-community add comm1
   set-med 50
   set-route-origin igp
   policy-result permit
   ```

The routing policy is now defined and ready to be attached to one or more routing protocols.

Apply a Routing Policy to BGP

A routing policy takes effect only after you attach it to a routing protocol. BGP supports applying policies on import (route reception) and export (route advertisement).

Use the following procedure to attach the routing policy to BGP import and export control points.

1. **configure terminal**

   Enter global configuration mode.

2. **vrf** *vrf-name*

   Enter the VRF that contains the BGP instance that you want to modify.

3. **router bgp**

   Enter BGP configuration mode.

4. **address-family ipv4 unicast**

   Select the routing context (such as IPv4 unicast or IPv6 unicast) to apply policies to the address family.

5. **import-policy** *policy-name*

   Apply the routing policy to inbound BGP routes. The BGP process evaluates this policy before adding received routes to the RIB.

6. **export-policy** *policy-name*

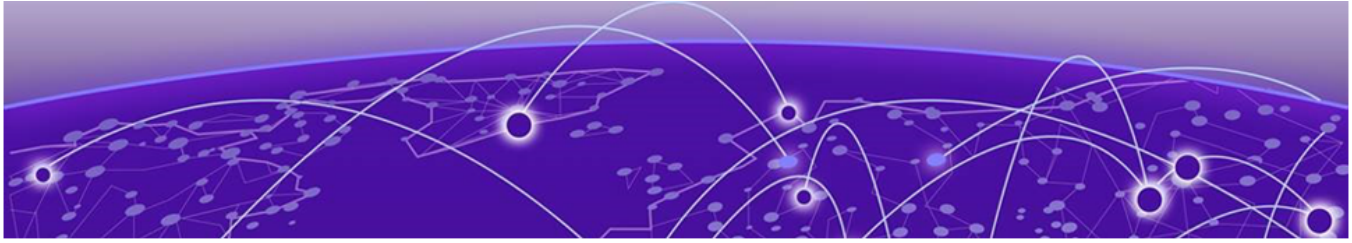
   Apply the routing policy to outbound BGP advertisements. This lets you control which routes are advertised and how their attributes appear to neighbors.

7. **exit**

   Exit address family configuration mode.

```
device# show running-config vrf tenant1
 vrf tenant1
  router bgp
   address-family ipv4 unicast
    import-policy map1
    export-policy map1
```

The routing policy is now actively controlling BGP route import and export behavior.



Key Chain Management

Key Chain Management Overview

Keychain management creates and maintain sequences of keys for secure communication with peers.

- Scaleability:
  - Supports up to 128 keychains.
  - Supports up to eight keys per keychain.
  - Allows configuration of authentication tolerance.
- Key Rollover: Provides a mechanism to manage key rollover using send and receive lifetimes.
- Purpose: Maintains secure communication for data plane and control plane traffic.

CLIs for Keychain Management

To configure a keychain:

For details about the commands in this section, see the *Extreme OS ONE SR Command Reference Guide.*

1. Enter global configuration mode:

   ```
   device# configure terminal
   ```

2. Create a keychain and enter keychain configuration mode:

   ```
   device(config)# keychain key1
   ```

3. Configure the tolerance value to extend the validity of an expired key during rollover. You can specify either an integer from 0 (the command default) to 600 seconds or the **forever** keyword.

   ```
   device(config-keychain-key1)# tolerance 400
   ```

4. Specify the key ID for the keychain:

   ```
   device(config-keychain-key1)# key-id 4
   ```

5. Configure the cryptographic algorithm for the key in the keychain. You can specify one of the following keywords: { **aes_128_cmac_96** | **crypto_none** | **hmac_md5** | **hmac_sha_1** | **hmac_sha_1_12** | **hmac_sha_1_20** | **hmac_sha_1_96** | **hmac_sha_256** | **md5** | **sha_1** }

```
device(config-keychain-key1-key-4)# crypto md5
```

6. Configure the receive lifetime of the key. Use the **start-time** keyword to specify a time (expressed in CCYY-MM-DDTHH:MM:SS) after which the key becomes valid to use. You can optionally use the **end-time** keyword to specify a time (expressed in CCYY-MM-DDTHH:MM:SS) after which the key can no longer be used.

```
device(config-keychain-key1-key-4)# receive-lifetime start-time 2022-05-30T20:20:00
end-time 2022-05-31T11:00:00
```

7. Configure the secret for the key to authenticate packets:

```
device(config-keychain-key1-key-4)# secret-key mysecret
```

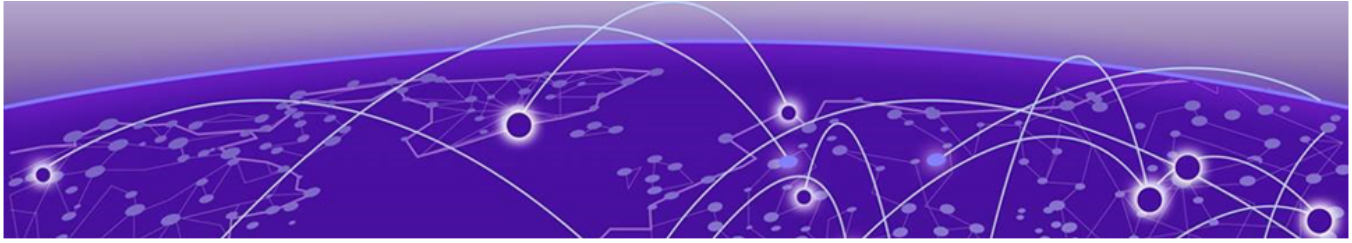
8. Configure the send lifetime of the key. Use the **start-time** keyword to specify a time (expressed in CCYY-MM-DDTHH:MM:SS) after which the key becomes valid to use. You can optionally use the **end-time** keyword to specify a time (expressed in CCYY-MM-DDTHH:MM:SS) after which the key can no longer be used. You can optionally use the send-receive { **true** | **false** } command: When send-receive is set to true (the default), the send lifetime can be used in the receive direction. When set to false, the configuration becomes asymmetric.

```
device(config-keychain-key1-key-4)# send-lifetime start-time 2022-05-31T11:00:00 end-
time 2022-05-31T11:00:10 send-receive true
```

9. (Optional) Confirm the keychain configuration.

```
device(config-keychain-key1-key-4)# do show running-config keychain key1

keychain key1
  tolerance 400
  key-id 4
    secret-key-
hashed e292kK4qXs32az2yrfkTQKaA2hq4JWF5a4UE25QkdpZzaYFKo1BQ3YaDiEjBunA0UPL14hsKjkz/
aQuFWuW9jA==
    crypto md5
    send-lifetime start-time 2022-05-31T11:00:00 end-time 2022-05-31T11:00:10 send-
receive true
    receive-lifetime start-time 2022-05-30T20:20:00 end-time 2022-05-31T11:00:00
  !
  key-id 33
!
device(config-keychain-key1-key-4)#
```



Management Security

Use this chapter to learn about the system-wide minimum TLS version configuration feature in Extreme OS ONE.

TLS Minimum Version Support

The TLS Minimum Version Support feature enhances security and administrative control. This lets administrators enforce a minimum TLS version (TLS 1.2 or TLS 1.3) across all TLS-enabled applications using CLI and gNMI. By default, the global minimum TLS version for all services is TLS 1.2.

Key Features

- **Global Configuration**: Set a minimum TLS version that applies to all client and server applications using TLS.
- **Flexibility**: Exclude specific applications from the enforced minimum TLS version to ensure compatibility with legacy systems or third-party systems that don't support newer TLS versions.
- **Multiple Application Exceptions**: Configure multiple applications with different TLS minimum versions, handled as a list.

Services Impacted by TLS Minimum Version Configuration

The TLS minimum version configuration affects the following services:

- Client: Syslog, LDAP, RADIUS, HTTPS
- Server: gRPC

Each service requires specific configuration commands to apply the TLS minimum version settings, such as importing SSL profiles and configuring service-specific settings.

Syslog Client Configuration with TLS

When the Syslog client is configured with the `secure-forwarding tls` option, it uses TLS for secure log forwarding.

The following is an example configuration command:

- Import the SSL profile.
```
certificate-manager import ssl-profile-id sp1 ca-certificate protocol scp host
192.0.2.0 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
```
- Associate the TLS profile with the remote server.
```
tls-profile-id sp1
```
- Verify the configuration: show running-config system logging remote-server.
```
show running-config system logging remote-server
system
  logging
    remote-server 192.0.2.0
      secure-forwarding tls
      mode-transport tcp
      remote-port 525
      tls-profile-id sp1
    !
  !
!
```

gRPC Server Configuration

The TLS minimum version is applied only when the instance is explicitly enabled or disabled. Existing connections won't be affected, but new connections established after the enable or disable operation will follow the configured TLS minimum version.

The following is an example configuration command:

- Generate an SSL profile.
```
certificate-manager generate ssl-profile-id ssl-reserved-generated
certificate-extension san 192.0.2.0
```
- Import a CA certificate.
```
certificate-manager import ssl-profile-id
ssl-reserved-generated ca-certificate protocol scp host 192.0.2.0 certificate
/tmp/cert.crt user user1 password pass1 vrf mgmt-vrf
```
- Configure the gRPC server.
```
certificate-id ssl-reserved-generated
```
- (Optional) Verify the configuration.
```
show running-config system grpc-server
system
  grpc-server DEFAULT
    certificate-id ssl-reserved-generated
    port 443
    enable
  !
!
```

LDAP Client Configuration with LDAPS and TLS Minimum Version

When the LDAP client is set up with the LDAPS option, it uses TLS for secure connections. If a minimum TLS version is configured, all new connections adhere to this setting.

The following is an example configuration command:

- Import the SSL profile using the **`certificate-manager`** command, specifying the SSL profile ID, CA certificate, and other details like host, certificate path, username, password, and VRF.

```
certificate-manager import ssl-profile-id sp1 ca-certificate protocol scp host
192.0.2.0 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
```

- Associate the SSL profile with the LDAP server using the **`ssl-profile-id`** command under the LDAP server configuration mode..

```
ssl-profile-id sp1
```

- (Optional) Verify the configuration by checking the running configuration for the AAA server group LDAP.

```
show running-config system aaa server-group ldap
system
aaa
   server-group ldap
     server 192.0.2.0
       base-dn example.com
       ldaps
       ssl-profile-id sp1
     !
   !
!
!
```

LDAP Authentication

LDAPS authentication occurs during SSH login and gNMI user authentication. It uses the configured minimum TLS version for these attempts.

RADIUS Client Configuration with RADSEC and TLS Minimum Version

When the RADIUS client is configured with the `RADSEC` option, it uses TLS for secure connections. If a minimum TLS version is configured, new connections will adhere to this setting.

The following is an example configuration command:

- Import the SSL profile using the **`certificate-manager`** command, specifying details such as the SSL profile ID, CA certificate, host, certificate path, username, password, and VRF.

```
certificate-manager import ssl-profile-id sp1 ca-certificate protocol scp host
192.0.2.0 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
```

- Associate the SSL profile with the RADIUS server by using the **`ssl-profile-id`** command under the RADIUS server configuration mode.

```
ssl-profile-id sp1
```

- (Optional) Verify the configuration by checking the running configuration for the AAA server group RADIUS.

```
show running-config system aaa server-group radius
system
  aaa
    server-group radius
      server 192.0.2.0
        secret-key-hashed QSARezGQul4kBEcysLCaqe1Q6xVncFq8v6eEMaTgqWsRUu1/
SSWWaxyCMl4YaoEA5pLm0vy2cCVydlgg0lx+ng==
        radsec
        ssl-profile-id sp1
      !
    !
  !
```

RADSEC Authentication

RADSEC authentication is used during SSH login and gNMI user authentication. It applies the configured minimum TLS version for these attempts.

HTTPS Client Configuration for Firmware Download

When you download firmware using the HTTPS option, the transfer occurs over TLS, and the configured minimum TLS version is applied.

Importing the SSL Profile

To enable secure firmware downloads, import the SSL profile using the **`certificate-manager`** command with the following details:

- **ssl-profile-id**: Use ssl-reserved-https for firmware updates.
- **ca-certificate**: Specify the protocol (SCP or SFTP), host, certificate file, username, password, and optional source IP address and VRF name.

The ssl-reserved-https SSL profile stores the CA certificate required for secure firmware updates using HTTPS.

The following is an example configuration command:

```
certificate-manager import ssl-profile-id ssl-reserved-https ca-certificate protocol scp
host<remote-ip> certificate <certificate-file> user <remote-user>password <remote-user-
password> [source-ip address] [vrf vrf-name]

system firmware update https://<url>/firmware.bin

system firmware fullinstall https://<url>/firmware.bin
```

CLI Commands for Minimum TLS Version

- Use the **`tls`** command to enter TLS system configuration (config-system-tls) mode.
- Use the **`service`** command to specify a TLS-enabled client/server service and enter TLS service-level system configuration (config-system-tls-service-*`name`*) mode.
- Use the **`min-version`** (TLS system configuration) command to set the global TLS minimum version for all TLS-enabled services on a device.

- Use the **min-version** (TLS service-level system configuration) command to set the TLS required minimum version for a specific TLS-enabled application on a device.
- Use the **show tls min-version** command to display the minimum version of Transport Layer Security (TLS) used by the services that are running on a device.

For information on syntax and command examples, see the *Extreme OS ONE SR Command Reference Guide*.

YANG Data Model

The YANG data model for the new CLI is designed to configure transport security, specifically, TLS minimum version settings.

```
+--rw ex-sys-transport-security:transport-security
     |  +--rw ex-sys-transport-security:tls
     |     +--rw ex-sys-transport-security:config
     |     |  +--rw ex-sys-transport-security:min-version?   identityref
     |     +--ro ex-sys-transport-security:state
     |     |  +--ro ex-sys-transport-security:min-version?   identityref
     |     +--rw ex-sys-transport-security:exceptions
     |        +--rw ex-sys-transport-security:exception* [service]
     |           +--rw ex-sys-transport-security:service    -> ../config/service
     |           +--rw ex-sys-transport-security:config
     |           |  +--rw ex-sys-transport-security:service?       identityref
     |           |  +--rw ex-sys-transport-security:min-version?   identityref
     |           +--ro ex-sys-transport-security:state
     |              +--ro ex-sys-transport-security:service?       identityref
     |              +--ro ex-sys-transport-security:min-version?   identityref
```

Event Log Messages

```
{
       "LogID": 7039,
        "Details": {
            "Level": "Info",
            "Msg": "TLS minimum version configuration applied successfully.",
            "Cause": "Indicates that the TLS minimum version command was configured
successfully.",
            "Remedy": "No action required.",
            "Impact": "All TLS-enabled services will use the configured minimum version
during the handshake."
    }
    },
    {
    "LogID": 7040,
    "Details": {
        "Level": "Info",
        "Msg": "TLS minimum version configuration removed successfully.",
        "Cause": "Indicates that the TLS minimum version command was removed
successfully.",
        "Remedy": "No action required.",
        "Impact": "All TLS-enabled services will revert to using the default minimum
version during the handshake."
    }
    },
    {
    "LogID": 7041,
```

```
    "Details": {
        "Level": "Error",
        "Msg": "TLS minimum version configuration failed.",
        "Cause": "Indicates that the TLS minimum version command could not be
configured.",
        "Remedy": "No action required.",
        "Impact": "All TLS-enabled services will continue using the default minimum
version during the handshake."
    }
    },
    {
    "LogID": 7042,
    "Details": {
        "Level": "Info",
        "Msg": "Service-specific TLS minimum version configuration applied successfully.",
        "Cause": "Indicates that a service-specific TLS minimum version command was
configured successfully.",
        "Remedy": "No action required.",
        "Impact": "The specified service will use its configured minimum TLS version
during the handshake, overriding the global setting."
    }
    },

  {

 "LogID": 7043,
    "Details": {
        "Level": "Info",
        "Msg": "Service-specific TLS minimum version configuration removed successfully.",
        "Cause": "Indicates that the service-specific TLS minimum version command was
removed successfully.",
        "Remedy": "No action required.",
        "Impact": "The service will fall back to using the global minimum TLS version
during the handshake."
    }
    },
    {
    "LogID": 7044,
    "Details": {
        "Level": "Error",
        "Msg": "Service-specific TLS minimum version configuration failed.",
        "Cause": "Indicates that the service-specific TLS minimum version command could
not be configured.",
        "Remedy": "No action required.",
        "Impact": "The service will fall back to using the global minimum TLS version
during the handshake."
    }
    }
```


ICMP Rate Limiting on the Management Interface

ICMP (Internet Control Message Protocol) is widely used for diagnostic and control purposes such as ping (echo-request/echo-reply) operations. While essential for network troubleshooting, unrestricted ICMP traffic can lead to Denial of Service (DoS) attacks, network congestion, and resource exhaustion.

To mitigate these risks, an ICMP rate limit on the management interface ensures that the system controls the number of ICMP packets that it processes within a defined threshold. This prevents abuse while maintaining functionality for legitimate diagnostics.

ICMP Rate Limiting on the Management Interface Overview

The ICMP rate limiting feature lets you control and monitor ICMP echo-request (ping) traffic on the management interface. This protects the device from ping flooding attacks and excessive ICMP traffic that could impact device performance.

This feature ensures device manageability by enabling necessary diagnostics while dropping excessive traffic. It specifically throttles ICMP traffic to let you troubleshoot safely without affecting management traffic such as SSH or HTTPS.

This feature provides configuration commands to apply ICMP iptables rules for restricting the number of pings on the management interface. It also provides a **`show`** command to display the aggregated counters of IPv4 and IPv6 ICMP packets that the management interface accepts or rejects.

This feature applies only to out-of-band and in-band (RME) ICMP traffic on the management interface.

Key Capabilities

The implementation of this feature delivers three core functionalities.

Configuration

The ICMP rate limiting feature on the management interface has the following core configuration functionality:

- Rate limit control—You can configure the maximum rate of ICMP echo-request packets that you can permit per second on the management interface.
- Dual stack support—You can configure independent rate limiting for both IPv4 and IPv6 ICMP traffic.
- Default protection—The system automatically applies a default rate limit of one packet per second.
- Persistent configuration—The system saves and stores rate limit settings across reboots.

Monitoring

The ICMP rate limiting feature on the management interface has the following core monitoring functionality:

- Real-time statistics—You can vew live counters showing Accepted Packets (Number of ICMP echo-request packets within the rate limit) and Dropped Packets (Number of ICMP echo-request packets exceeding the rate limit)
- Per-protocol visibility—You can view separate counters for IPv4 and IPv6 ICMP traffic.
- Counter clearing—You can clear counters via the **`clear counter interface management 0`** command, which includes the ICMP counters.

Default Behavior

The ICMP rate limiting feature on the management interface has the following core default behavior:

- The system enables ICMP rate limiting by default with a rate of one packet per second for both IPv4 and IPv6.
- The system silently drops all ICMP echo-request packets exceeding the configured rate.

Configure ICMP Rate Limiting on the Management Interface

You can configure the ICMP rate limit to a value other than the default rate of one packet per second for both IPv4 and IPv6. To configure the ICMP rate limit, complete the following steps.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter the mode for configuring the management interface.

```
device(config)# interface management 0
```

   In this release, the only supported management interface name is 0.

3. Specify the IPv4 rate limit in pings per second.

```
device(config-if-mgmt-0)# ipv4 icmp ratelimit 2
```

   The range is 2 to 1000.

4. Specify the IPv6 rate limit in pings per second.

```
device(config-if-mgmt-0)# ipv6 icmp ratelimit 2
```

   The range is 2 to 1000.

5. (Optional) Verify the ICMP rate limiting configuration.

```
device(config-if-mgmt-0)# do show running-config interface management 0

interface management 0
  ipv4 icmp ratelimit 2
device#device(config-if-mgmt-0)#
```

The following example configures ICMP rate limiting on the management interface. This example limits the IPv4 and IPv6 ICMP rates to two pings per second.

```
device# configure terminal
device(config)# interface management 0
device(config-if-mgmt-0)# no ipv4 address dhcp
device(config-if-mgmt-0)# no ipv6 address dhcp
device(config-if-mgmt-0)# ipv4 address 192.0.2.0/24
device(config-if-mgmt-0)# ipv6 address 2001:db8:f000:1005:10:38:212:20/64
device(config-if-mgmt-0)# ipv4 icmp rate limit 2
device(config-if-mgmt-0)# ipv6 icmp ratelimit 2
device(config-if-mgmt-0)# no shutdown
device(config-if-mgmt-0)#
```

The following example displays the configuration of the management interface that is currently running on the device. This example sets the IPv4 and IPv6 ICMP rate limits to two pings per second.

```
device# show running-config interface management 0

interface management 0
  no ipv4 address dhcp
  no ipv6 address dhcp
  ipv4 address 192.0.2.0/24
  ipv6 address 2001:db8:f000:1005:10:38:212:20/64
  ipv4 icmp ratelimit 2
  ipv6 icmp ratelimit 2
  no shutdown
device#
```

Monitor ICMP Rate Limiting on the Management Interface

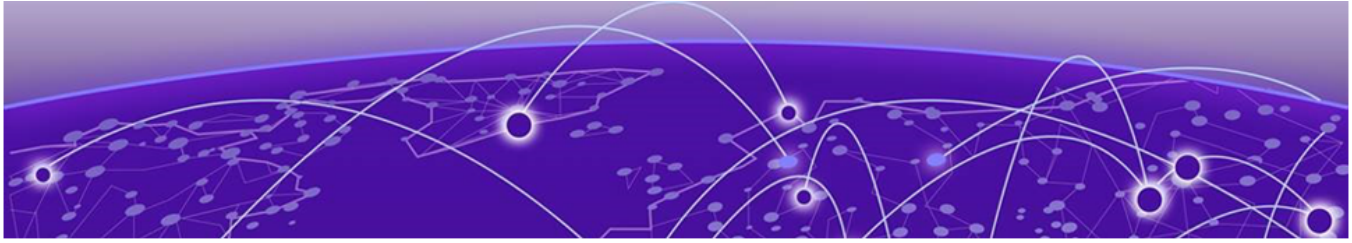
You can display the aggregated ICMP accepted and dropped counters for IPv4 and IPv6 on the management interface. To monitor ICMP management interface counters, use the **`show counters icmp interface management 0`** command. Examine the following statistics:

- ICMPv4 accepted packets
- ICMPv4 dropped packets
- ICMPv6 accepted packets
- ICMPv6 dropped packets

The following example shows how to monitor ICMP management interface counters.

```
device# show counters icmp interface management 0

ICMP Packet Summary:
  ICMPv4 Accepted Packets: 8
  ICMPV4 Dropped Packets:  11
  ICMPv6 Accepted Packets: 0
  ICMPV6 Dropped Packets:  0
device#
```



gNSI Certificate Management

Use this topic to learn about the gNSI certificate management, such as managing and associating SSL profile, validating token, monitoring certificates, and the migration procedure.

gNSI Certificate Management Overview

gNSI (gRPC Network Security Interface) is a set of gRPC-based services that provide a standardized way to manage network security configurations and operations on devices. It facilitates certificate management within network devices by enabling secure communication using TLS/SSL certificates.

The gNSI Certz service lets a client to replace an application certificate, CA certificate, or some combination of these artifacts on the device, providing improved certificate management capabilities with granular control over individual certificates and certificate authorities.

You can share an SSL profile across the applications.

gNSI Certz Service Remote Procedure Calls (RPC)

The gNSI Certz service defines the following RPCs for SSL profile management:

- AddProfile(): Adds a new SSL profile to the device with empty artifacts (certificate, CA certificate). The client must then populate the artifacts using the Rotate RPC. Duplicate profile names are rejected with an error.
- DeleteProfile(): Removes an existing SSL profile.
- GetProfileList(): Retrieves a list of SSL profile IDs on the device.
- Rotate(): Replaces existing certificate, CA certificate, or both in an SSL profile
- GetCertificates(): Fetches certificate artifacts for a specified SSL profile. This is a custom RPC.

The following is a logical view of the artifacts managed by gNSI Certz service available in certz.proto:

```
Target (as seen from gNSI.certificate microservice point of view)
|
+-+ SSL profile for gNXI; always present and immutable;
  | ssl_profile_id := "system_default_profile"
| |
| +-+ certificate
| | +- certificate (with public key)
| | +- private key
| |
| +-+ trust bundle (Certificate Authority certificates)
| | +- CA Root certificate
| | +- CA Intermediate Certificate
| |
+-+ Another SSL profile used by another service
  |
  +-+ certificate
  | +- certificate (with public key)
  | +- private key
  |
  +-+ trust bundle (Certificate Authority certificates)
  | +- CA Root certificate
  | +- CA Intermediate Certificate
  |
  ..
```

Configure Certificates

Follow this procedure to configure certificates.

1. Generate App Certificate

   Add the app certificates to a reserved SSL profile (ssl-reserved-generated).

   ```
   device# certificate-manager generate ssl-profile-id ssl-reserved-generated certificate-
   extension san 192.0.2.0
   Generated app certificate successfully for ssl-profile-id ssl-reserved-generated
   ```

2. Import App Certificate

   Use this command to copy certificate and (optional) private key from external remote server to the system certificates store. If private key is omitted, the imported certificate can only be used for token validation.

   ```
   device# certificate-manager import ssl-profile-id sp1 app-certificate protocol scp
   host 192.0.2.0 certificate /tmp/cert.pem key /tmp/key.pem user user1 password **** vrf
   mgmt-vrf
   Imported app certificate successfully to ssl-profile-id sp1

   device# certificate-manager import ssl-profile-id sp1 app-certificate protocol scp
   host 192.0.2.0 certificate /tmp/cert.pem user user1 password **** vrf mgmt-vrf
   Warning: Importing app-cert without key. This certificate cannot be used for tls
   handshake, it can be only used for token validation
   Imported app certificate successfully to ssl-profile-id sp1
   ```

3. Show App Certificate

   Use this command to display app certificate that is included in the specified SSL profile. When 'all' option is chosen, app certificates for all SSL profiles are shown.

   ```
   device# show certificate-manager app-certificate ssl-profile-id sp1
   App level certificates:
   certificate-id: sp1
   Endpoints using this certifcate-id:[type:EP_DAEMON  endpoint:"grpc-server DEFAULT"]
   ```

```
sha256
Fingerprint=AF:21:61:D0:15:A3:32:07:7D:11:C2:D8:E6:85:61:8B:43:A2:52:C1:95:54:BE:5B:0F:
CC:93:9D:DE:E3:23:BC
subject=C=US, ST=CA, O=Extreme Networks, OU=Extreme OS ONE switching and Routing,
CN=extremenetworks.com
issuer=C=US, ST=CA, O=Extreme Networks, OU=Extreme OS ONE switching and Routing,
CN=extremenetworks.com
notBefore=Mar 19 06:12:42 2025 GMT
notAfter=Mar 17 06:12:42 2035 GMT


device# show certificate-manager app-certificate all
App level certificates:
certificate-id:sp1
Endpoints using this certifcate-id:[type:EP_DAEMON  endpoint:"grpc-server DEFAULT"]
sha256
Fingerprint=AF:21:61:D0:15:A3:32:07:7D:11:C2:D8:E6:85:61:8B:43:A2:52:C1:95:54:BE:5B:0F:
CC:93:9D:DE:E3:23:BC
subject=C=US, ST=CA, O=Extreme Networks, OU=Extreme OS ONE switching and Routing,
CN=extremenetworks.com
issuer=C=US, ST=CA, O=Extreme Networks, OU=Extreme OS ONE switching and Routing,
CN=extremenetworks.com
notBefore=Mar 19 06:12:42 2025 GMT
notAfter=Mar 17 06:12:42 2035 GMT

certificate-id:sp2
Endpoints using this certifcate-id:[type:EP_DAEMON  endpoint:"token-validator DEFAULT"]
sha256
Fingerprint=AF:21:61:D0:15:A3:32:07:7D:11:C2:D8:E6:85:61:8B:43:A2:52:C1:95:54:BE:5B:0F:
CC:93:9D:DE:E3:23:BC
subject=C=US, ST=CA, O=Extreme Networks, OU=Extreme OS ONE switching and Routing,
CN=extremenetworks.com
issuer=C=US, ST=CA, O=Extreme Networks, OU=Extreme OS ONE switching and Routing,
CN=extremenetworks.com
notBefore=Mar 19 06:12:42 2025 GMT
notAfter=Mar 17 06:12:42 2035 GMT
```

4. Import CA Certificate

   Use this command to copy trusted CA certificates from external remote server to system trust certificates list.

```
device# certificate-manager import ssl-profile-id sp1 ca-certificate protocol scp host
192.0.2.0 certificate /tmp/cert.crt user user1 password **** vrf mgmt-vrf
Imported CA certificate successfully to ssl-profile-id sp1
```

5. Export CA certificate

   Use this command to copy the system default trusted CA certificates to an external remote server to establish GNMI or GNOI connection.

```
device# certificate-manager export ca-certificate default protocol scp remote-server
192.0.2.0 remote-file /tmp/cert.pem user user1 password **** vrf mgmt-vrf
Exported switch 'default' CA certificate successfully
```

6. Show CA Certificate

   Use this command to display CA certificate that is included in the specified SSL profile. When 'all' option is chosen, CA certificates for all SSL profiles are shown.

```
device# show certificate-manager ca-certificate ssl-profile-id sp2
CA certificates:
certificate-id: sp2
Endpoints using this certifcate-id:[type:EP_DAEMON  endpoint:"token-validator DEFAULT,
server-group radius 192.0.2.0"]
sha256
FingerPrint=AF:21:61:D0:15:A3:32:07:7D:11:C2:D8:E6:85:61:8B:43:A2:52:C1:95:54:BE:5B:0F:
CC:93:9D:DE:E3:23:BC
```

```
subject=CN=extremenetworks.com,OU=Extreme OS ONE switching and Routing,O=Extreme
Networks,ST=CA,C=US
issuer=CN=extremenetworks.com,OU=Extreme OS ONE switching and Routing,O=Extreme
Networks,ST=CA,C=US
notBefore=Mar 19 06:12:42 2025 UTC
notAfter=Mar 17 06:12:42 2035 UTC


devce# show certificate-manager ca-certificate all
CA certificates:
certificate-id:sp1
Endpoints using this certifcate-id:[type:EP_DAEMON  endpoint:"grpc-server DEFAULT"]
sha256
FingerPrint=AF:21:61:D0:15:A3:32:07:7D:11:C2:D8:E6:85:61:8B:43:A2:52:C1:95:54:BE:5B:0F:
CC:93:9D:DE:E3:23:BC
subject=CN=extremenetworks.com,OU=Extreme OS ONE switching and Routing,O=Extreme
Networks,ST=CA,C=US
issuer=CN=extremenetworks.com,OU=Extreme OS ONE switching and Routing,O=Extreme
Networks,ST=CA,C=US
notBefore=Mar 19 06:12:42 2025 UTC
notAfter=Mar 17 06:12:42 2035 UTC

certificate-id:sp2
Endpoints using this certifcate-id:[type:EP_DAEMON  endpoint:"token-validator DEFAULT,
server-group radius 1.1.1.1"]
sha256
FingerPrint=AF:21:61:D0:15:A3:32:07:7D:11:C2:D8:E6:85:61:8B:43:A2:52:C1:95:54:BE:5B:0F:
CC:93:9D:DE:E3:23:BC
subject=CN=extremenetworks.com,OU=Extreme OS ONE switching and Routing,O=Extreme
Networks,ST=CA,C=US
issuer=CN=extremenetworks.com,OU=Extreme OS ONE switching and Routing,O=Extreme
Networks,ST=CA,C=US
notBefore=Mar 19 06:12:42 2025 UTC
notAfter=Mar 17 06:12:42 2035 UTC
```

7. Import PKCS Certificate

   Use this command to copy PKCS certificate key bundle from external remote server to the system certificates store.

```
device# certificate-manager import-pkcs ssl-profile-id sp1 app-certificate protocol
scp host 192.0.2.0 file /tmp/cert.pkcs12 passphrase **** user user1 password **** vrf
mgmt-vrf
Imported app certificate successfully to ssl-profile-id sp1
```

8. Delete Certificate

   Use this command to delete app certificate and ca-certificate that are included in the specified SSL profile. When 'all' option is chosen, app certificate and CA certificate for all SSL profiles are deleted.

```
device# certificate-manager delete ssl-profile-id sp1
Deleted ssl-profile-id sp1 successfully.

device# certificate-manager delete all
Deleted all ssl-profile-ids successfully.
```

SSL Profile Management

SSL profiles are containers that include various security artifacts, such as application certificates (with public key and private key) and CA (Certificate Authority) trust bundles. They are created under gNSI Certz service model.

To use the certificates, the applications must associate with a SSL profile. You can associate each SSL profile with multiple applications, so that you can manage certificates efficiently across services.

Maximum SSL Profiles

The device supports a maximum of 64 SSL profiles, which is sufficient to accommodate the maximum instances required by various applications, including the following instance types:

- gRPC: 32 instances
- LDAP: 6 instances
- RADIUS: 6 instances
- Syslog: 10 instances
- Token Validator: 1 instance

Reserved SSL Profiles

The device maintains certain SSL profiles for system operations that require certificates. These profiles are prefixed with "ssl-reserved" and can be deleted by the user. The following reserved SSL profiles are available:

- ssl-reserved-generated: Stores the application certificate generated using the **`certificate-manager generate`** command, used by the gRPC server instance.

  ```
  certificate-manager generate ssl-profile-id ssl-reserved-generated certificate-extension san <ip-addr>
  ```

- ssl-reserved-ztp: Stores the CA certificate downloaded during the secure Zero-Touch Provisioning (ZTP) workflow.
- ssl-reserved-https: Stores the CA certificate necessary for firmware updates and copy operations using HTTPS. You can import the necessary CA certificates to this profile via CLI command.

Associate SSL Profile

To use imported certificates, an application instance must associate an SSL profile with itself. Any changes to the SSL profile association, such as dissociation or updates, must be handled by the application. If the application cannot handle these changes gracefully, a restart may be required.

The device supports the following profile associations:

- gRPC Server: Associates with SSL profile through existing certificate-id attribute.
- LDAP: Yang data model is augmented to include an SSL profile to enable LDAP client instance association.
- RADIUS: Yang data model is augmented to include an SSL profile to enable RADIUS client instance association.

- Syslog: Client instance associates with SSL profile through the existing tls-profile-id attribute.
- Token Validator: Yang data model is augmented to include an SSL profile to enable Token Validator instance association.

1. gRPC Server Configuration

   Associates with an SSL profile using the certificate-id attribute. The profile must contain the gRPC server certificate and CA certificate (for mutual authentication).
   To associate an SSL profile with a gRPC server instance, run the following command:

```
device(config)# system
device(config-system)# grpc-server <instance-name> (if no name specified, Default
instance will be created)
device(config-system-grpc-server-DEFAULT)# certificate-id <ssl-reserved-generated>
device(config-system-grpc-server-DEFAULT)# enable
```

   The following is an example CLI of gPRC server configuration:

```
device# show running-config system grpc-server
system
  grpc-server DEFAULT
    certificate-id ssl-reserved-generated
    port 443
    enable
  !
!
```

   On updating the SSL profile, gRPC server instance continues to use the previous certificate until restarted. It should be restarted using the following command:

```
device(config-system-grpc-server-DEFAULT)# no enable
device(config-system-grpc-server-DEFAULT)# enable
```

2. LDAP Configuration

   Yang data model is augmented to include SSL profile for LDAP client instance association. The profile should contain the required CA certificate to validate the LDAP server certificate.
   To configure LDAP with SSL profile association, run the following command:

```
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# server-group ldap
device(config-system-aaa-server-group-ldap)# server 192.0.2.0
device(config-system-aaa-server-group-ldap-server-192.0.2.0)# ssl-profile-id sp1
device(config-system-aaa-server-group-ldap-server)# ldaps
```

   The following is an example CLI output to import a certificate and associate with configured ssl profile:

```
device# certificate-manager import ssl-profile-id sp1 ca-certificate protocol scp host
192.0.2.0 certificate /tmp/cert.crt user vikhanna password **** vrf mgmt-vrf

device# show running-config system aaa server-group ldap
system
  aaa
    server-group ldap
      server 192.0.2.0
        base-dn example.com
        ldaps
        ssl-profile-id sp1
      !
    !
  !
!
```

There is no impact if the SSL profile is updated. They use the latest certificate during authentication attempts.

3. RADIUS Configuration

   Yang data model is augmented to include SSL profile for RADIUS client instance association. The profile must contain the required CA certificate to validate the RADIUS server certificate.

   To configure RADIUS with SSL profile association, run the following command:

```
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# server-group radius
device(config-system-aaa-server-group-radius)# server 192.0.2.0
device(config-system-aaa-server-group-radius-server-192.0.2.0)# ssl-profile-id sp1
device(config-system-aaa-server-group-radius-server-192.0.2.0)# radsec
```

   The following is an example CLI to import a certificate and associate with configured ssl profile:

```
device# certificate-manager import ssl-profile-id sp1 ca-certificate protocol scp host
192.0.2.0 certificate /tmp/cert.crt user vikhanna password **** vrf mgmt-vrf

device# show running-config system aaa server-group radius
system
  aaa
    server-group radius
      server 192.0.2.0
        secret-key-hashed QSARezGQul4kBEcysLCaqe1Q6xVncFq8v6eEMaTgqWsRUu1/
SSWWaxyCMl4YaoEA5pLm0vy2cCVydlgg0lx+ng==
        radsec
        ssl-profile-id sp1
      !
    !
  !
!
```

   There is no impact if the SSL profile is updated. They use the latest certificate during authentication attempts.

4. Syslog Configuration

   Client instance associates with SSL profile using the `tls-profile-id` attribute.

   The profile must contain the required CA certificate to validate the Syslog server certificate.

   To configure secure syslog with SSL profile, run the following command:

```
device(config)# system
device(config-system)# logging
device(config-system-logging)# remote-server 192.0.2.0
device(config-system-logging-remote-server-192.0.2.0)# secure-forwarding tls
ddevice(config-system-logging-remote-server-192.0.2.0)# tls-profile-id sp1
```

   The following is an example CLI to import a certificate and associate with configured ssl profile:

```
device# certificate-manager import ssl-profile-id sp1 ca-certificate protocol scp host
192.0.2.0 certificate /tmp/cert.crt user vikhanna password **** vrf mgmt-vrf

device# show running-config system logging remote-server
system
  logging
    remote-server 192.0.2.0
      secure-forwarding tls
      mode-transport tcp
      remote-port 525
      tls-profile-id sp1
    !
```

```
  !
!
```

Any change in the Syslog configuration results in a restart of the syslogd daemon, except when the associated SSL profile is updated with a new CA certificate or deleted, which requires a manual restart.

5. Token Validator Configuration

   The YANG data model is augmented to include SSL profile for Token Validator instance association. The profile should contain the required certificate to validate the JWT token.
   To configure the token validator with SSL profile association, run the following command:

```
device# certificate-manager import ssl-profile-id sp1 app-certificate protocol scp
host 192.0.2.0 certificate /tmp/cert.pem user user1 password **** vrf mgmt-vrf
Warning: Importing app-cert without key. This certificate cannot be used for tls
handshake, it can be only used for token validation
Imported app certificate successfully to ssl-profile-id sp1

device# show running-config system aaa token-validator
system
  aaa
    token-validator DEFAULT
      ssl-profile-id sp1
    !
  !
!
```

   There is no impact if the SSL profile is updated. The latest certificate is used to validate the JWT token.

Token Validation Configuration

Token validation enables authentication of external users (users not authenticated on the device). These users are authenticated by an external entity, which signs a JWT (JSON Web Token) token included in gNMI requests, with a private key. The corresponding public key certificate must be imported on the device for successful token validation.

The following are the key features of token validator:

- Token Validator configuration includes association with an SSL profile containing the necessary certificate.
- Only one token validator instance can be configured.
- For incoming gNMI requests with a bearer token, the device iterates through each token validator and stops when token validation is successful.
- If all token validators fail, the token validation logic falls back to validating device-generated tokens for backward compatibility.
- On successful validation, the username is extracted from the JWT claim and included in configuration and security audit logs.

Token Validator Configuration and Data Model

The Token Validator configuration is part of the openconfig-system module, with the following data model:

```
module: openconfig-system
  +--rw system
    +--rw aaa
      +--rw extr-aaa-token-ext:token-validators
        +--rw extr-aaa-token-ext:token-validator* [name]
          +--rw extr-aaa-token-ext:name -> ../config/name
          +--rw extr-aaa-token-ext:config
            | +--rw extr-aaa-token-ext:name? string
            | +--rw extr-aaa-token-ext:ssl-profile-id? string
            | +--rw extr-aaa-token-ext:type? enumeration
```

The following is an example configuration of token validator:

```
device# show running-config system aaa token-validator
system
  aaa
    token-validator DEFAULT
      ssl-profile-id sp1
    !
  !
!
```

JWT Token Requirements

The JWT token claims must include the following attributes, with role and sub being particularly important:

- role: Subject (username)
- sub: User role (for example, admin and user)
- Other attributes as needed (for example, org, ver, id, requestor, iss, exp, nbf, iat, and jti)

Audit Logs

Successful token validation results in configuration and security audit logs with the extracted username.

Configuration Audit Log

```
2025-04-13 13:54:07.022 UTC +0000 LogID:6001 info Msg: xco-user/192.0.2.0/http/grpc
Method:/gnmi.gNMI/Set Status:OK
Request:update:{path:{elem:{name:"keychains"}  elem:{name:"keychain"  key:{key:"name"
value:"authnewone"}}  elem:{name:"config"}  elem:{name:"name"}}  val:
{string_val:"authnewone"}}
```

Security Audit Log

```
2025-04-13 13:54:07.017 UTC +0000 LogID:7002 info Msg: User authentication is
successful for user: xco-user
```

Monitor Certificates

The device continuously monitors the validity of certificates in use, including both application and CA certificates. It tracks SSL profiles associated with various

applications, such as gRPC server, LDAP, RADIUS, Syslog, and Token Validator. When displaying certificates via CLI, the device shows the endpoints using each certificate.

1. Display certificates

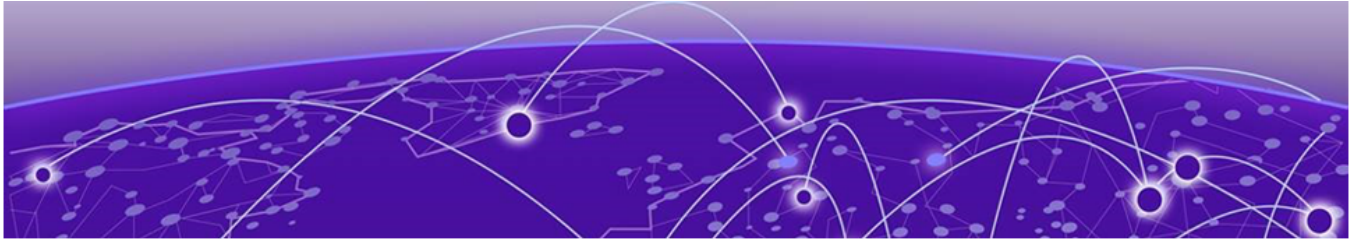
   To view the certificates used by each device, run the following command:
   The command displays the certificate details, its validity period, and the applications using the certificate. When multiple applications share the same SSL profile, the CLI output might look like this:

```
device# show certificate-manager ca-certificates sp1
CA certificates:
 certificate-id: sp1
Endpoints using this certificate-id:[type:EP_DAEMON  endpoint:"token-validator
DEFAULT, server-group ldap 192.0.2.0, server-group radius 192.0.2.1, logging remote-
server 192.0.2.2"]
```

2. Enable certificate expiry alerts

   Administrators can enable certificate expiry alerts by configuring syslog log level notifications based on the number of days left before certificate expiry.
   For details on certificate expiry alerts and the configuration procedure, see:

   - Certificate Expiry Alert on page 104
   - Configure Certificate Expiry Alert on page 105



Mutual Authentication

Use this chapter to learn how to configure mutual TLS (mTLS) for the Remote Procedure Calls (gRPC) server as well as for services such as *LDAP*, *RADIUS*, and remote system logging (*SYSLOG*). This chapter also describes how to configure mTLS for the HTTPS client for firmware operations.

Mutual Authentication Overview

Mutual authentication provides mutual TLS (mTLS) support for the Remote Procedure Calls (gRPC) server as well as critical services such as *LDAP*, *RADIUS*, and remote system logging (*SYSLOG*). It also provides mTLS support for the HTTPS client that you use for system firmware operations. Mutual authentication uses a unified configuration model across these capabilities to let you enable mTLS via the **`mutual-tls`** command and associate certificate profiles that the gNSI certificate infrastructure manages.

Extreme OS ONE supports both server-side and client-side roles. It operates as a server for secure gNMI services (via its gRPC server). It also functions as a client for services such as LDAP, RADIUS, and SYSLOG and acts as an HTTPS client for firmware download and downgrade operations.

Extreme OS ONE mTLS client authentication on server instances uses an X.509 certificate. mTLS enhances the security of client-server communications in Extreme OS ONE by enabling two-way certificate based authentication. Unlike traditional TLS, which authenticates only the server, mTLS ensures that both the client and the server will verify each other's identities.

For secure connections, Extreme OS ONE lets you import application certificates for the LDAP, RADIUS, and SYSLOG services. You can also import a root CA certificate for the Remote Procedure Calls (gRPC) server.

Configure Mutual Authentication for gRPC

Note
You must disable the Remote Procedure Calls (gRPC) server instance before altering its configuration. Then, you must re-enable the instance to restore its functionality. To disable or enable a gRPC server instance, you use the **enable** (gRPC server configuration) command. For details, see the *Extreme OS ONE SR Command Reference Guide*

To configure mutual authentication for the gRPC server, complete the following steps.

1. Import the gRPC Certificate Authority (CA) and application certificates.
```
device# certificate-manager import ssl-profile-id grpc_test app-certificate protocol
scp host 192.0.2.0 certificate /tmp/client.pem key /tmp/client.key user user password
**** vrf mgmt-vrf
device# certificate-manager import ssl-profile-id grpc_test ca-certificate protocol
scp host 192.0.2.0 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
```
2. Enter global configuration mode.
```
device# configure terminal
```
3. Enter system configuration mode.
```
device(config)# system
```
4. Specify the name of a gRPC server and enter gRPC server system configuration mode.
```
device(config-system)# grpc-server mygrpcserver
```
5. Enable mutual TLS (mTLS) client authentication on the gRPC server.
```
device(config-system-grpc-server-mygrpcserver)# mutual-tls
```
6. (Optional) Verify the mutual authentication configuration.
```
device(config-system-grpc-server-mygrpcserver)# do show running-config system grpc-
server

system
  grpc-server mygrpcserver
    mutual-tls
!
device(config-system-grpc-server-mygrpcserver)#
```

The following example configures a gRPC server instance. This example enables mTLS client authentication on the instance:

```
device# configure terminal
device(config)# system
device(config-system)# grpc-server mygrpcserver
device(config-system-grpc-server-mygrpcserver)# certificate-id test_cert_id
device(config-system-grpc-server-mygrpcserver)# port 9340
device(config-system-grpc-server-mygrpcserver)# mutual-tls
device(config-system-grpc-server-mygrpcserver)# vrf vrf-blue
device(config-system-grpc-server-mygrpcserver)# enable
device(config-system-grpc-server-mygrpcserver)#
```

The following example displays the gRPC server instance configuration that is running currently on the device. This example enables mTLS client authentication on the instance:

```
device# show running-config system grpc-server
```

```
system
  grpc-server mygrpcserver
   certificate-id test_cert_id
   port 9340
   mutual-tls
   vrf vrf-blue
   enable
!
device#
```

Configure Mutual Authentication for LDAP

To configure mutual authentication for LDAP, complete the following steps.

1. Import your LDAP Certificate Authority (CA) and application certificates.

```
device# certificate-manager import ssl-profile-id ldap_test app-certificate protocol
scp host 198.51.100.7 certificate /tmp/client.pem key /tmp/client.key user user
password **** vrf mgmt-vrf
device# certificate-manager import ssl-profile-id ldap_test ca-certificate protocol
scp host 198.51.100.7 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
```

2. Enter global configuration mode.

```
device# configure terminal
```

3. Enter system configuration mode.

```
device(config)# system
```

4. Enter AAA system configuration mode.

```
device(config-system)# aaa
```

5. Enter the mode for configuring an LDAP server group.

```
device(config-system-aaa)# server-group ldap
```

6. Specify the IPv4 or IPv6 address of an LDAP server instance in the LDAP server group and enter AAA LDAP server group server system configuration mode.

```
device(config-system-aaa-server-group-ldap)# server 192.0.4.0
```

7. Enable mutual TLS (mTLS) client authentication on the LDAP server instance.

```
device(config-system-aaa-server-group-ldap-server-192.0.4.0)# mutual-tls
```

8. (Optional) Verify the mutual authentication configuration.

```
device(config-system-aaa-server-group-ldap-server-192.0.4.0)# do show running-config
system aaa server-group ldap

system
  aaa
    server-group ldap
      server 192.0.4.0
        mutual-tls
      !
    !
  !
 !
 device(config-system-aaa-server-group-ldap-server-192.0.4.0)#
```

The following example configures an LDAP server instance. In this example, mTLS client authentication is enabled on the instance:

```
device# configure terminal
device(config)# system
```

```
device(config-system)# aaa
device(config-system-aaa)# server-group ldap
device(config-system-aaa-server-group-ldap)# server 192.0.4.0
device(config-system-aaa-server-group-ldap-server-192.0.4.0)# base-dn example.com
device(config-system-aaa-server-group-ldap-server-192.0.4.0)# ldaps
device(config-system-aaa-server-group-ldap-server-192.0.4.0)# source-interface ethernet
0/1
device(config-system-aaa-server-group-ldap-server-192.0.4.0)# ssl-profile-id ldap_test
device(config-system-aaa-server-group-ldap-server-192.0.4.0)# vrf vrf1
device(config-system-aaa-server-group-ldap-server-192.0.4.0)# mutual-tls
device(config-system-aaa-server-group-ldap-server-192.0.4.0)#
```

The following example displays the LDAP server instance configuration that is running currently on the device. In this example, mTLS client authentication on the instance is enabled:

```
device# show running-config system aaa server-group ldap

system
  aaa
    server-group ldap
      server 192.0.4.0
        base-dn example.com
        ldaps
        source-interface ethernet  0/1
        ssl-profile-id ldap_test
        vrf vrf1
        mutual-tls
      !
    !
  !
 !
 device(config-system-aaa-server-group-ldap-server-192.0.4.0)#
```

Configure Mutual Authentication for RADIUS

To configure mutual authentication for RADIUS, complete the following steps.

1. Import your RADIUS Certificate Authority (CA) and application certificates.

   ```
   device# certificate-manager import ssl-profile-id radius_test ca-certificate protocol
   scp host 192.0.2.0 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
   device# certificate-manager import ssl-profile-id radius_test app-certificate protocol
   scp host 192.0.2.0 certificate /tmp/client.pem key /tmp/client.key user user password
   **** vrf mgmt-vrf
   ```

2. Enter global configuration mode.

   ```
   device# configure terminal
   ```

3. Enter system configuration mode.

   ```
   device(config)# system
   ```

4. Enter AAA system configuration mode.

   ```
   device(config-system)# aaa
   ```

5. Enter the mode for configuring a RADIUS server group.

   ```
   device(config-system-aaa)# server-group radius
   ```

6. Specify the IPv4 or IPv6 address of a RADIUS server instance in the RADIUS server group and enter AAA RADIUS server group server system configuration mode.

   ```
   device(config-system-aaa-server-group-radius)# server 192.0.2.0
   ```

7. Enable mutual TLS (mTLS) client authentication on the RADIUS server instance.

```
device(config-system-aaa-server-group-radius-server-192.0.2.0)# mutual-tls
```

8. (Optional) Verify the mutual authentication configuration.

```
device(config-system-aaa-server-group-radius-server-192.0.2.0)# do show running-config
system aaa server-group radius

system
  aaa
    server-group radius
      server 192.0.2.0
        mutual-tls
      !
    !
  !
!
device(config-system-aaa-server-group-radius-server-192.0.2.0)#
```

The following example configures a RADIUS server instance. In this example, mTLS client authentication is enabled on the instance:

```
device# configure terminal
device(config)# system
device(config-system)# aaa
device(config-system-aaa)# server-group radius
device(config-system-aaa-server-group-radius)# server 192.0.2.0
device(config-system-aaa-server-group-radius-server-192.0.2.0)# radsec
device(config-system-aaa-server-group-radius-server-192.0.2.0)# secret-key sharedsecret
device(config-system-aaa-server-group-radius-server-192.0.2.0)# source-interface ethernet
0/2
device(config-system-aaa-server-group-radius-server-192.0.2.0)# ssl-profile-id radius_test
device(config-system-aaa-server-group-radius-server-192.0.2.0)# vrf vrf2
device(config-system-aaa-server-group-radius-server-192.0.2.0)# mutual-tls
device(config-system-aaa-server-group-radius-server-192.0.2.0)#
```

The following example displays the RADIUS server instance configuration that is running currently on the device. In this example, mTLS client authentication on the instance is enabled:

```
device# show running-config system aaa server-group radius

system
  aaa
    server-group radius
      server 192.0.2.0
        radsec
        secret-key sharedsecret
        source-interface ethernet 0/2
        ssl-profile-id radius_test
        vrf vrf2
        mutual-tls
      !
    !
  !
!
device#
```

Configure Mutual Authentication for SYSLOG

To configure mutual authentication for a remote system logging (SYSLOG) server, complete the following steps.

Note
For the given profile ID associated with the SYSLOG client, the absence of a valid application certificate or CA certificate causes an error.

1. Import your SYSLOG Certificate Authority (CA) and application certificates.

```
device# certificate-manager import ssl-profile-id syslog_test ca-certificate protocol
scp host 192.0.2.0 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
device# certificate-manager import ssl-profile-id syslog_test app-certificate protocol
scp host 192.0.2.0 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
password ****
```

2. Enter global configuration mode.

```
device# configure terminal
```

3. Enter system configuration mode.

```
device(config)# system
```

4. Enter logging system configuration mode.

```
device(config-system)# logging
```

5. Specify the IPv4 or IPv6 address of the SYSLOG server and enter remote logging server system configuration mode.

```
device(config-system-logging)# remote-server 192.0.2.0
```

6. Specify the SSL profile for TLS encryption for securely forwarding system logs to the remote SYSLOG server.

```
device(config-system-logging-remote-server-192.0.2.0)# tls-profile-id syslog_test
```

7. Enable mutual TLS (mTLS) client authentication on the remote SYSLOG server.

```
device(config-system-logging-remote-server-192.0.2.0)# mutual-tls
```

8. (Optional) Verify the mutual authentication configuration.

```
device(config-system-logging-remote-server-192.0.2.0)# do show running-config system
logging remote-server

system
  logging
    remote-server 192.0.2.0
      tls-profile-id syslog_test
      mutual-tls
    !
  !
!
device(config-system-logging-remote-server-192.0.2.0)#
```

The following example configures a logging remote server. This example enables mTLS client authentication on the server:

```
device# configure terminal
device(config)# system
device(config-system)# logging
device(config-system-logging)# remote-server 192.0.2.0
device(config-system-logging-remote-server-192.0.2.0)# mode-transport relp
device(config-system-logging-remote-server-192.0.2.0)# remote-port 525
```

```
device(config-system-logging-remote-server-192.0.2.0)# secure-forwarding tls
device(config-system-logging-remote-server-192.0.2.0)# source-interface ethernet 0/3
device(config-system-logging-remote-server-192.0.2.0)# tls-profile-id syslog_test
device(config-system-logging-remote-server-192.0.2.0)# vrf mgmt-vrf
device(config-system-logging-remote-server-192.0.2.0)# mutual-tls
Warning: Existing Host configuration changed
device(config-system-logging-remote-server-192.0.2.0)#
```

The following example displays the logging remote server configuration that is running currently on the device. This example enables mTLS client authentication on the server:

```
device# show running-config system logging remote-server

system
  logging
    remote-server 192.0.2.0
      mode-transport relp
      remote-port 525
      secure-forwarding tls
      source-interface ethernet 0/3
      tls-profile-id syslog_test
      vrf mgmt-vrf
      mutual-tls
    !
  !
!
device#
```

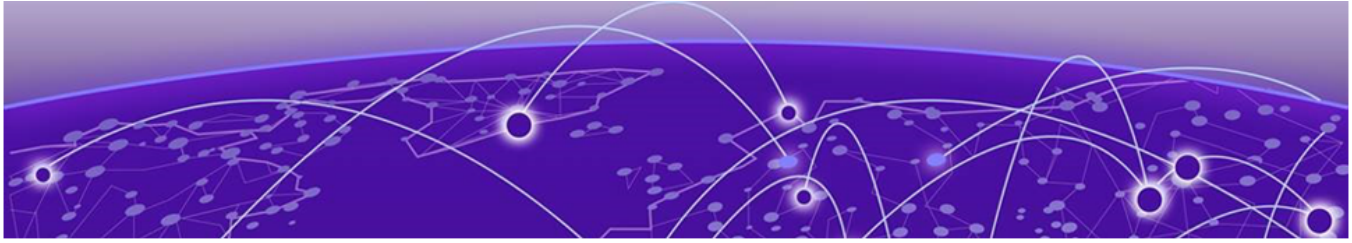
Configure Mutual Authentication for HTTPS

Any firmware download via HTTPS will occur using TLS. Extreme OS ONE enables mutual TLS (mTLS) authentication automatically based on certificate presence. For example, after you import your certificates for HTTPS, the following command triggers mTLS:

```
device# system firmware
update https://myartifacts1.mydomain.com:8081/artifactory/local-snapshots/SR/22.2.0.0/
build/ExtremeOneSR-22.2.2.0-098/install/ExtremeOneSR-22.2.2.0-098.bin
```

To enable mTLS for HTTPS, use the **`ssl-profile-id ssl-reserved-https`** keyword and variable combination of the **`certificate-manager import`** command to import your HTTPS Certificate Authority (CA) and application certificates to your SSL profile. The following example imports these certificates to an SSL profile named ssl-reserved-https.

```
device# certificate-manager import ssl-profile-id ssl-reserved-https ca-certificate
protocol scp host 192.0.2.0 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
device# certificate-manager import ssl-profile-id ssl-reserved-https app-certificate
protocol scp host 192.0.2.0 certificate /tmp/cert.crt user user password **** vrf mgmt-vrf
```



Certificate Expiry Alert

Certificate Expiry Alert

All cryptographic certificates have an effective lifetime. This lifetime is defined in the validity fields `notBefore` and `notAfter` values stored within each cryptographic certificate. Ideally, a cryptographic certificate should not be used prior to the date configured in the *notBefore* field. The cryptographic certificate is considered as *expired* beyond the date configured in the *notAfter* field and should not be used after that date.

When a cryptographic certificate nears its expiration date, then a notification is generated with the configured warning level.

Notifications can be RASLog or SNMP or both.

Notifications to users can be classified as *Warning* or *Error* as seen in the RASLOG entries. Messages of the type *Warnings* are only generated if the alert levels are configured. The valid alert levels are INFO, MINOR, MAJOR, and CRITICAL and are configured independent of each other. These classifications are applicable to both RASLOG entries and SNMP Notifications.

The notifications of the type *Error* are always generated irrespective of the configured alert levels. By default, RASLOGs are always written for notifying certificate expiry. SNMP notifications are only generated when SNMP is enabled on the device.

For the *Warning* type of messages, when notifications are generated, these incorporate the configured alert level, along with the details of the expiring certificate. This is generated for each certificate that will expire in the near term.

A single warning is generated when the number of remaining days for a certificate's expiry is equal to the configured period for that severity level.

For the *Error* type of messages, notifications are always generated once a day at midnight (00:00 hours) for each certificate that has expired. This notification is generated till the expired certificates are renewed or their validity extended.

Depending on the value of the `notAfter` field in each certificate, the generation of the notification might be delayed by up to 24 hours.

Things to Note about Notifications for Certificate Management

- A single alert is issued if the number of remaining days until expiration is equal to the number of days configured for that expiry-level. To calculate the time remaining until a certificate expires, compare the certificate's expiry timestamp with the current timestamp, both measured to the second. The resulting time difference is then converted into the number of days remaining.
- Certificate validity verification is performed once every 24 hours at midnight (00:00 hours). When configured, the certificate expiration event might not get triggered immediately and depends on the time of day when the configuration is performed. It is only triggered when the device's clock next reaches 00:00 hours.
- If a certificate has expired, then, the notification is sent every 24 hours till the certificate is changed or its validity is extended. This notification is independent of the expiry-level configuration and does not contain any information about the alert level. Extreme OS ONE does not allow importing an already expired certificate.
- If the system time is manually changed after a notification is sent, Extreme OS ONE does not resend the same notification unless the specific expiry-level for which the notification is sent is reconfigured or the specific certificate for which the notification is sent is reimported.

Certificates Monitored for Expiry

The system actively monitors certificates on devices for their validity, including application and CA certificates. It tracks SSL profiles used by various applications, such as gRPC server, LDAP, RADIUS, Syslog, and Token Validator. When displaying certificates via CLI, the system shows associated endpoints.

If multiple applications share the same SSL profile, the CLI output will list all relevant endpoints. Administrators can configure certificate expiry alerts with customizable syslog log level notifications based on the number of days left before expiry. Use the **`expiry-alert`** command for the certificate expiry setup.

```
device# configure terminal
device(config)# system
device(config-system)# certificate-manager
device(config-system-cert-mgr)# expiry-alert
device(config-system-cert-mgr-exp)# critical 30
device(config-system-cert-mgr-exp)# major 60
device(config-system-cert-mgr-exp)# minor 80
device(config-system-cert-mgr-exp)# info 90
device(config-system-cert-mgr-exp)#
```

The system monitors application certificates and CA certificates for expiry.

Configure Certificate Expiry Alert

Certificate expiry alerts can be configured for four alert levels. These alert levels can be configured independent of each other. Use the **`expiry-alert`** command to enter

Certificate manager expiry alert system configuration (config-system-cert-mgr-exp) mode.

1. Enter the **configure terminal** mode.

```
device# configure terminal
device(config)# system
device(config-system)# certificate-manager
device(config-system-cert-mgr)# expiry-alert
```

2. Configure the *Info* certificate expiry alert level. Here the *Info* level is configured and set to ninety (90) days.

```
device(config-system-cert-mgr-exp)# info 90
device (config)#
```

3. Configure the *Minor* certificate expiry alert level. Here the *Minor* level is configured and set to eighty (80) days.

```
device(config-system-cert-mgr-exp)# minor 80
device  (config)#
```

4. Configure the *Major* certificate expiry alert level. Here the *Major* level is configured and set to sixty (60) days.

```
device(config-system-cert-mgr-exp)# major 60
device (config)#
```

5. Configure the *Critical* certificate expiry alert level. Here the *Critical* level is configured to thirty (30) days.

```
device(config-system-cert-mgr-exp)# critical 30
device (config)#
```

The certificate expiry alert level is configured for the *Info*, *Minor*, *Major*, and *Critical* levels only.

The notifications are generated in the following order, based on the preceding configuration example:

- On the 90th day, you will receive one *Warning* notification with the level *info*.
- On the 80th day, you will receive one *Warning* notification with the level *minor*. You will not receive any notifications of the level *info* in between.
- On the 60th day from certificate expiry, you will receive one *Warning* notification with the level *major*. You will not receive any notifications of the level *minor* in between.
- On the 30th day from certificate expiry, you will receive one *Warning* notification with the level *critical*. You will not receive any notifications of the level *major* in between.
- Once the certificate expires, you will receive an *Error* notification every day at midnight (00:00 hours) until the certificate is renewed or its validity is extended.

Each *Warning* notification will be sent with the alert level mentioned in the message and the details of the certificate that is about to expire. The calculation, as to when to send the notification, will consider time to the granularity of days and will disregard the hours, minutes, or seconds remaining till certificate expiry.

Certificate validity verification is performed once every 24 hours at midnight (00:00 hours). When configured, the certificate expiration event might not get triggered

immediately, and it depends on the time of day when the configuration is performed. It is only triggered when the device's clock next reaches 00:00 hours.

Notifications will be sent once the configuration is done. When the system's clock is reset within the last 24 hours to the previous day, the certificate expiry alert will not be generated.