Extreme ONE OS Switching v22.2.0.0 Security Configuration Guide

GRUB Protection, ACLs, AAA, and Certificate Management

9039346-00 Rev AA
July 2025

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 Networksand 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/

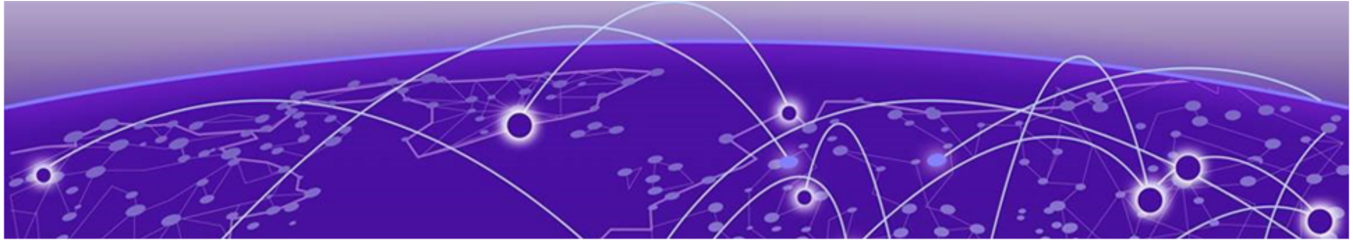
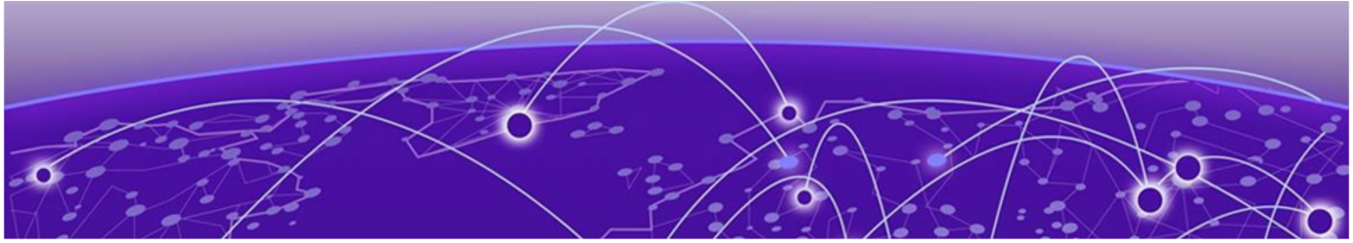
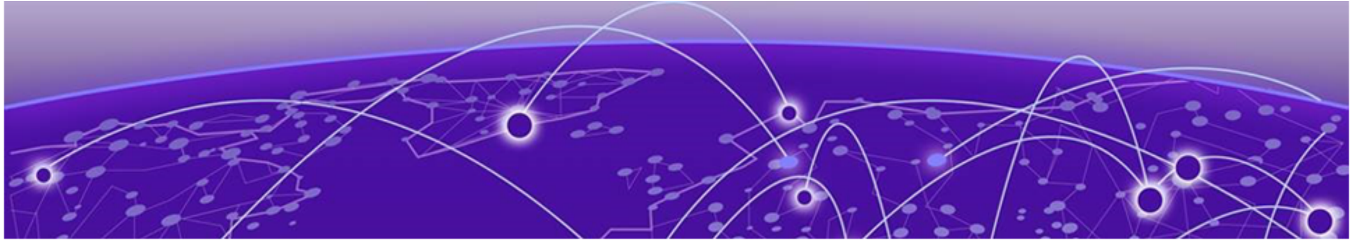


Table of Contents



Abstract

The Extreme ONE OS Switching v22.2.0.0 Security Configuration Guide provides implementation-level procedures for securing the Extreme 8730-32D platform. It covers GRUB bootloader protection with PBKDF-based password hashing, ACL configuration using OpenConfig YANG models and CLI, and classifier microservice integration with FWD HAL for hardware programming. The guide includes RACLs for control-plane filtering, SSH hardening with OpenSSH_9.6p1, and TLS 1.2/1.3 support. It details AAA integration with TACACS+, LDAP, and RADIUS, routing policy enforcement, and gNSI-based certificate lifecycle management, including SSL profile association, token validation, and expiry alerting.



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 or routers, the product is referred to as *the switch* or *the router*.

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.

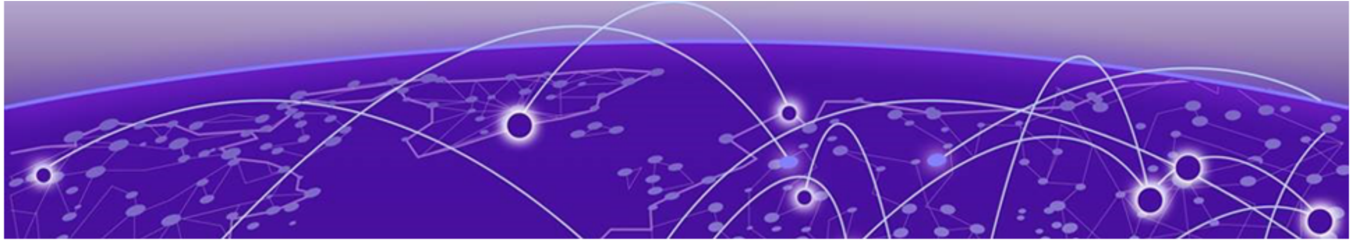
Send Feedback

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

This is the initial release of Extreme ONE OS Switching 22.2.0.0 software release.

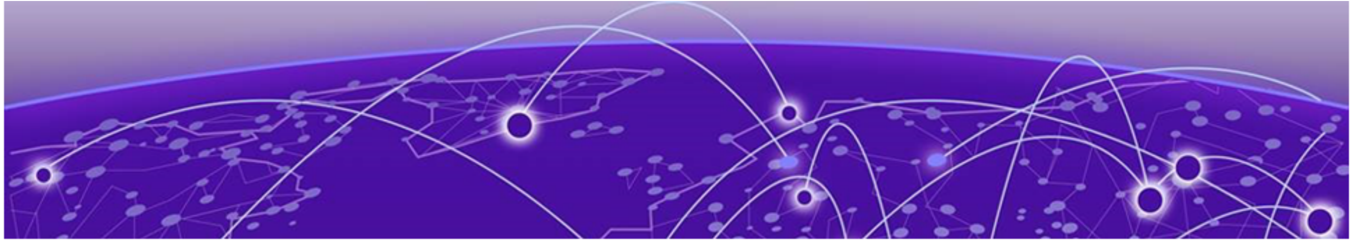
For additional information, refer to the *Extreme ONE OS Release Notes* for this version.

Supported Hardware

For instances in which a topic or part of a topic applies to some devices but not to others, the topic specifically identifies the devices.

Extreme ONE OS 22.2.0.0 supports Extreme 8730-32D 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.



Securing GRUB

Securing GRUB Boot Loader

GRUB's boot loader interface is accessible to anyone with console access, allowing you to edit boot menu entries, add or delete entries, or access the GRUB prompt. To secure this feature, a default username and password are provided to protect the GRUB menu. Additionally, Extreme ONE OS offers a CLI and API for users to modify these credentials, restricting access to the boot loader menu.

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

- Include a default Grub username and password in Extreme ONE OS for protecting Grub boot loader menu.
- Enforce user to change this default credential (username and/or password) when user logs into CLI, with option to use same password as admin user password.
- Allow user to change this default username or password via CLI and GNMI.
- Allow user to configure Grub username/password using a ZTP configuration file during initial provisioning.
- Generate a warning message during every boot if the default GRUB credentials have not been changed.
- Any inputs other than selecting the default boot option in Grub menu (attempt to boot from other boot entries, edit a boot entry, or enter GRUB command line interface) will require authentication.

Configuring GRUB Boot Loader Credentials using CLI

You can set up GRUB credentials to protect the GRUB boot loader by running the following command as an admin user:

```
switch(config-system-grub)# username root password <password>
```

Ensure that the username must start with alpha-numeric or underscore characters, and only contain alpha-numeric, underscore, or period characters.

Ensure that the plain-text password must satisfy password strength requirements of password-attributes(under aaa authentication password-attributes).

Key points:

- Only one user is supported for GRUB protection, with a default username 'root'.
- Credentials can be changed via CLI or GNMI.
- On first login, users are prompted to change the default password.
- The GRUB password can be set separately or synced with the admin user password.
- Passwords can be provided in plain text or as a hash generated.
- The boot loader configuration, including the username and password hash, is stored in `cdb` and appended to `grub.cfg`.

  GRUB Authentication

  - GRUB requires authentication for non-default boot entries, editing, or command line access.
  - The default boot entry (Open Network Linux) is unrestricted, allowing boot without authentication.
  - The same credentials protect all GRUB menu entries, including ONIE and Diag options.

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

Configuring GRUB Boot Loader Credentials using gNMI Command

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

Configuring GRUB Boot Loader Credentials using Copy Config Command

Using Default Config Copy

When you run 'copy default-config running-config' or perform a factory reset, the GRUB user/password in grub.cfg will revert to defaults before the device reboots. After the reboot, you'll need to use the default credentials to access the GRUB menu (except for default boot).

Using User Config Copy

If you copy a user config, 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 ZTP: Modify GRUB credentials using the GNMI path system/grub/config in the ZTP configuration file. If defaults aren't modified, an audit log will be generated on every boot.
- 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 ONE OS using ONIE will recreate default username/password in the config database and /mnt/onl/boot/grub.cfg.

Special Boot Modes

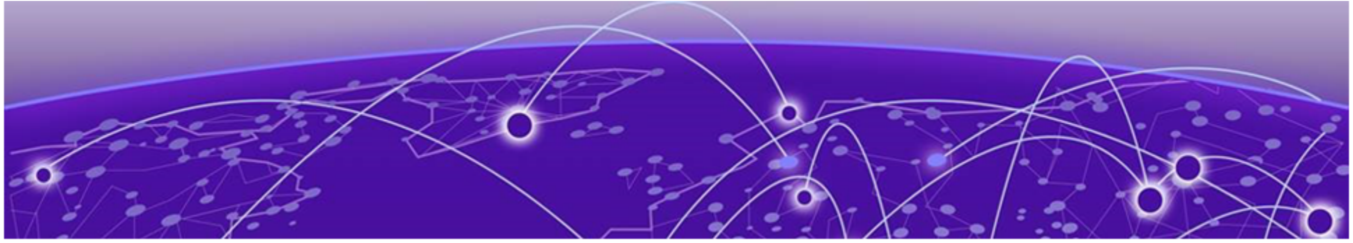
- Diag Boot: The system internal run diag CLI allows booting into diagnostics mode without requiring GRUB password. After diag run completion, password protection will be restored. There will be no change in the username and password.
- Full Install: The system firmware fullinstall CLI allows booting into ONIE mode without requiring GRUB password. After a full install, the device will have a new Extreme ONE OS installation with default username/password for GRUB.

Lost Password

The device will boot into Extreme ONE OS without needing a GRUB password. If the username/password is lost or forgotten, it can be changed using CLI.

Security

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



ACLs

Access Control List (ACL) Overview

An Access Control List (ACL) is a list of rules, known as Access Control Entries (ACEs), that can be attached to any classifier feature. An ACL has no effect unless it is attached to a feature.

Key Characteristics

1. Sequence ID: Each ACE is indexed and uniquely identified by a sequence ID, which determines the priority of the entry. Lower sequence IDs have higher priority.
2. TCAM: ACLs use Ternary Content-Addressable Memory (TCAM) to filter packets. Each ACE is programmed as an entry in the TCAM.
3. Matching Logic: All entries in the TCAM are matched simultaneously, and the action from the highest priority entry (lowest sequence ID) is taken.

Types of ACLs

1. MAC ACL: Qualifies on MAC (Link Layer) fields in a packet's header, with optional metadata fields.
2. IPv4 ACL: Qualifies on IPv4 (Network Layer) fields in a packet's header, with optional metadata and Transport Layer (TCP/UDP) fields.
3. IPv6 ACL: Qualifies on IPv6 (Network Layer) fields in a packet's header, with optional metadata and Transport Layer (TCP/UDP) 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 a primary interface for configuring networking devices. Using the CLI code, you can define the CLI tokens. The commands enable users to configure and manage ACLs, including creating, deleting, and showing ACL configurations and statistics.

The following CLI commands are available:

1. Create or Delete ACL.

   ```
   device(config)# [no] (ip | ipv6 | mac) access-list <ACL_NAME>
   ```

2. Configure ACEs.

   - IPv4 ACE

     ```
     ddevice(config-ip-acl)# [no] [seq <SEQ_NO>] (permit | deny) (ip | tcp | udp | icmp
     | esp| <PROTOCOL_NO>)
           (any | <SADDR> <SADDR_MASK>) (any | <DADDR> <DADDR_MASK>)
           [sport <L4PORT_NO>] [dport <L4PORT_NO>] [dscp <DSCP>] [vlan <VLAN_ID>] [count]
     ```

   - IPv6 ACE

     ```
     device(config-ipv6-acl)# [no] [seq <SEQ_NO>] (permit | deny) (ip | tcp | udp |
     icmpv6 | esp | <PROTOCOL_NO>)
                                  (any | <SADDR> <SADDR_MASK>) (any | <DADDR>
     <DADDR_MASK>)
                                  [sport <PORT_NO>] [dport <PORT_NO>] [dscp <DSCP>]
     [vlan <VLAN_ID>] [count]
     ```

   - MAC ACE

     ```
     device(config-mac-acl)# [no] [seq <SEQ_NO>] (permit | deny) (any | <SADDR>
     <SADDR_MASK>)
                 (any | <DADDR> <DADDR_MASK>) [vlan <VLAN_ID>] [pcp <PCP>] [etype
     <ETHTYPE>] [count]
     ```

3. Show Commands.

   - To see configuration

     ```
     device# show running-config  [(ip | ipv6 | mac)] access-list (all | <NAME>)
     ```

   - To see state and statistics

     ```
     device# show [(ip | ipv6 | mac)] access-list (all | <NAME>)
     ```

YANG Model for ACL Configuration

The YANG model defines the structure for ACL configuration, 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 of the OpenConfig ACL YANG model is used to store ACLs.

Key Components

1. ACL Sets: The `/acl/acl-sets` branch stores an ACL, indexed by name and type.
2. ACE: The `acl-entry* [sequence-id]` branch stores an ACE, indexed by sequence ID.

YANG Tree

The YANG tree structure is as follows:

```
+-rw acl
 +-rw acl-sets
    +-rw acl-set* [name type]
       +-rw name
       +-rw type
       +-rw config
       |  +-rw name
       |  +-rw type
       |  +-rw description?   string
       +-ro state
       |  +-ro name
       |  +-ro type
       |  +-ro description
       +-rw acl-entries
          +-rw acl-entry* [sequence-id]
             +-rw sequence-id
             +-rw config
             |  +-rw sequence-id
             |  +-rw description
             +-ro state
             |  +-ro sequence-id
             |  +-ro description
             |  +-ro matched-packets
             |  +-ro matched-octets
             +-rw actions
             |  +-rw config
             |  |  +-rw forwarding-action
             |  |  +-rw log-action
             |  |  +-rw extr-acl-ext:count
             |  +-ro state
             |     +-ro forwarding-action
             |     +-ro log-action
             |     +-ro extr-acl-ext:count
             +-rw extr-acl-ipv4-ext:npb-acl-ipv4
             |  +-rw extr-acl-ipv4-ext:config
             |  |  +-rw extr-acl-ipv4-ext:source-ipv4
             |  |  +-rw extr-acl-ipv4-ext:source-ipv4-mask
             |  |  +-rw extr-acl-ipv4-ext:destination-ipv4
             |  |  +-rw extr-acl-ipv4-ext:destination-ipv4-mask
             |  |  +-rw extr-acl-ipv4-ext:dscp
             |  |  +-rw extr-acl-ipv4-ext:protocol
             |  |  +-rw extr-acl-ipv4-ext:vlan-tag
             |  |  +-rw extr-acl-ipv4-ext:network-id-type?
             |  |  +-rw extr-acl-ipv4-ext:network-id
             |  |  +-rw extr-acl-ipv4-ext:source-port
             |  |  +-rw extr-acl-ipv4-ext:destination-port
             |  |  +-rw extr-acl-ipv4-ext:tcp-flags
             |  +-ro extr-acl-ipv4-ext:state
             |     +-ro extr-acl-ipv4-ext:source-ipv4
             |     +-ro extr-acl-ipv4-ext:source-ipv4-mask
             |     +-ro extr-acl-ipv4-ext:destination-ipv4
             |     +-ro extr-acl-ipv4-ext:destination-ipv4-mask
             |     +-ro extr-acl-ipv4-ext:dscp
             |     +-ro extr-acl-ipv4-ext:protocol
             |     +-ro extr-acl-ipv4-ext:vlan-tag
             |     +-ro extr-acl-ipv4-ext:network-id-type
             |     +-ro extr-acl-ipv4-ext:network-id
             |     +-ro extr-acl-ipv4-ext:source-port
             |     +-ro extr-acl-ipv4-ext:destination-port
             |     +-ro extr-acl-ipv4-ext:tcp-flags
             +-rw extr-acl-ipv6-ext:npb-acl-ipv6
             |  +-rw extr-acl-ipv6-ext:config
             |  |  +-rw extr-acl-ipv6-ext:source-ipv6
             |  |  +-rw extr-acl-ipv6-ext:source-ipv6-mask
             |  |  +-rw extr-acl-ipv6-ext:destination-ipv6
             |  |  +-rw extr-acl-ipv6-ext:destination-ipv6-mask
             |  |  +-rw extr-acl-ipv6-ext:dscp
             |  |  +-rw extr-acl-ipv6-ext:protocol
             |  |  +-rw extr-acl-ipv6-ext:vlan-tag
             |  |  +-rw extr-acl-ipv6-ext:network-id-type
             |  |  +-rw extr-acl-ipv6-ext:network-id
             |  |  +-rw extr-acl-ipv6-ext:source-port
             |  |  +-rw extr-acl-ipv6-ext:destination-port
             |  |  +-rw extr-acl-ipv6-ext:tcp-flags
             |  +-ro extr-acl-ipv6-ext:state
             |     +-ro extr-acl-ipv6-ext:source-ipv6
             |     +-ro extr-acl-ipv6-ext:source-ipv6-mask
             |     +-ro extr-acl-ipv6-ext:destination-ipv6
             |     +-ro extr-acl-ipv6-ext:destination-ipv6-mask
             |     +-ro extr-acl-ipv6-ext:dscp
             |     +-ro extr-acl-ipv6-ext:protocol
             |     +-ro extr-acl-ipv6-ext:vlan-tag
             |     +-ro extr-acl-ipv6-ext:network-id-type
             |     +-ro extr-acl-ipv6-ext:network-id
             |     +-ro extr-acl-ipv6-ext:source-port
             |     +-ro extr-acl-ipv6-ext:destination-port
             |     +-ro extr-acl-ipv6-ext:tcp-flags
             +-rw extr-acl-mac-ext:npb-acl-mac
                +-rw extr-acl-mac-ext:config
                |  +-rw extr-acl-mac-ext:source-mac
                |  +-rw extr-acl-mac-ext:source-mac-mask
                |  +-rw extr-acl-mac-ext:destination-mac
                |  +-rw extr-acl-mac-ext:destination-mac-mask
                |  +-rw extr-acl-mac-ext:pcp
                |  +-rw extr-acl-mac-ext:ethertype
                |  +-rw extr-acl-mac-ext:network-id-type
                |  +-rw extr-acl-mac-ext:network-id
                |  +-rw extr-acl-mac-ext:vlan-tag
                +-ro extr-acl-mac-ext:state
                   +-ro extr-acl-mac-ext:source-mac
                   +-ro extr-acl-mac-ext:source-mac-mask
                   +-ro extr-acl-mac-ext:destination-mac
                   +-ro extr-acl-mac-ext:destination-mac-mask
                   +-ro extr-acl-mac-ext:pcp
                   +-ro extr-acl-mac-ext:ethertype
                   +-ro extr-acl-mac-ext:network-id-type
                   +-ro extr-acl-mac-ext:network-id
                   +-ro extr-acl-mac-ext:vlan-tag
```

Statistics

Statistics per ACE in an ACL are collected from hardware and updated to SDB in the `matched-packets` and `matched-octets` fields in the YANG model. These statistics can be retrieved using **`GNMI get`** or **`show ACL`** command in CLI.

Attachment Points

An attachment point specifies where an ACL should work. The following attachment points are supported:

1. Physical Interface: Apply ACL to filter packets going through a physical port (for example, ethernet 0/1).
2. LAG (Link Aggregation Group): Apply ACL to filter packets going through a LAG (port-channel).
3. Note: If a physical port is added to a LAG group, an ACL cannot be attached to the physical port.
4. VE (Virtual Ethernet): Apply ACL to filter traffic flowing through a bridge domain.

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

5. Control-Plane: Apply ACLs for packets going to the CPU of the device (RACLs).
6. Breakout Port: Apply ACLs to breakout ports, which are treated like normal physical ports.

> **Note**
>
> - If an ACL is attached to an attachment point, the rules will be programmed to hardware regardless of whether the attachment point is admin/ operational up or down.
> - If a LAG or VE is deleted, the ACL applied on it will be removed automatically.
> - If a physical port is converted to a breakout port or vice versa, the ACLs on that port will be removed by the system.

Attachment Direction

The attachment direction specifies the direction in which an ACL is applied. 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:

1. Ingress: Filter packets coming into the switch.
2. Egress: Filter packets going out of the switch.

Security ACL

Security ACL is a feature in classifiers that selectively allows clients to access network resources.

How it Works

To configure a security ACL, follow these steps:

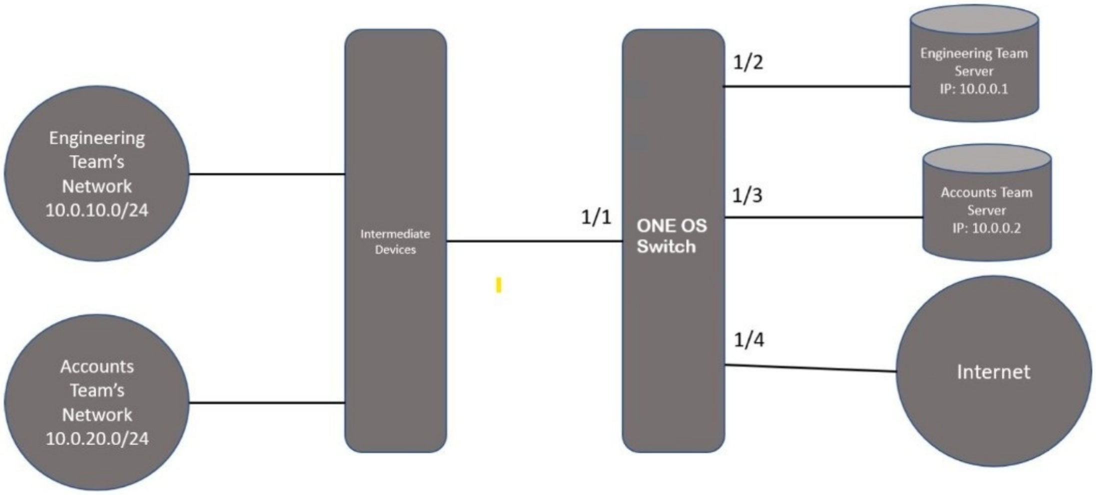
1. Create an ACL: Define an ACL with specific rules to filter traffic.
2. Attach the ACL: Attach the ACL to an attachment point (for example, interface) in the required direction (ingress or egress).

Example Use Case

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

- Engineering team: 10.0.10.0/24 network
- Accounts team: 10.0.20.0/24 network

And the goal is to prevent the Engineering team from accessing the Accounts team's server and vice versa, while allowing all teams to access the internet. Use Security ACL to achieve the desired traffic filtering and access control.


Engineering Team's Network 10.0.10.0/24
Accounts Team's Network 10.0.20.0/24
Intermediate Devices
1/1
ONE OS Switch
1/2
1/3
1/4
Engineering Team Server IP: 10.0.0.1
Accounts Team Server IP: 10.0.0.2
Internet

In the following two examples, the ACLs filter traffic based on specific rules, allowing or denying access to certain resources.

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 10.0.10.0/24 10.0.0.1
  seq 20 deny ipv4 any 10.0.0.1
  seq 30 permit ipv4 10.0.20.0/24 10.0.0.2
  seq 40 deny ipv4 any 10.0.0.2
interface ethernet 1/1
  ipv4 access-list ipAcl in
```

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 10.0.10.0/24 any
  seq 20 deny ipv4 any any
ipv4 access-list ipAclAccounts
```

```
  seq 10 permit ipv4 10.0.20.0/24 any
  seq 20 deny ipv4 any any
interface ethernet 1/2
  ipv4 access-list ipAclEngineering out
interface ethernet 1/3
  ipv4 access-list ipv4AclAccounts out
```

CLI Configuration Commands

To attach an ACL as a security ACL, use the following command:

```
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. Interface Table: Indexed by interface ID (for example, "ethernet 0/1", "ve 10", "port-channel 1").
2. Ingress ACL Sets: Attach ACLs to an interface for ingress traffic.
3. 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
      +--rw egress-acl-sets
        +--rw egress-acl-set* [set-name type]
          +--rw set-name
          +--rw type
          +--rw config
          |  +--rw set-name
          |  +--rw type
```

Config 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.

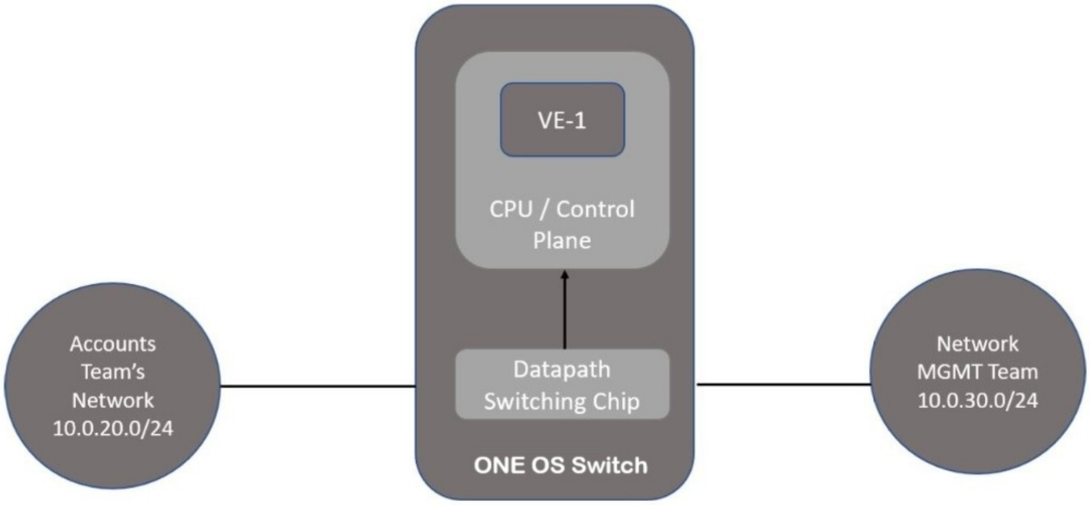
Receive ACL (RACL)

RACL, also known as Control Plane ACL, filters packets destined to IP interfaces configured on the 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. Global application: RACLs are applied globally to the device, not specific to any interface, LAG, VE, or VRF.
3. IP ACL only: 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.


VE-1
CPU / Control Plane
Datapath Switching Chip
ONE OS Switch
Accounts Team's Network 10.0.20.0/24
Network MGMT Team 10.0.30.0/24

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 10.0.30.0/24 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>
```

Config Validators

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

User Account and Password Configuration

Force Password Change At First Login

For enhanced security, change the default password immediately. No CLI configuration is needed. The system will automatically prompt the default admin user to update its password upon first successful login after:

- ONIE install
- Factory reset
- Full install (without preserving settings)
- Copy default config

Note
Force password change is enabled by default.

Force Password Age Out

To increase security, it is recommended that password for all accounts be changed frequently. This section describes how to force users, including admin users, to change their passwords on expiry of a pre-configured time interval. This is a global configuration.

Perform the following steps to force change of password on expiry of a pre-configured time interval.

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

```
device# configure terminal
```

4. Configure the setting to enforce changing of password after expiry of a set time period in days. This time duration is called *Age Out* duration.

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

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

Password Expiry Alert

By default, all the local users, excluding the 'admin-user' account, will receive password expiry alerts. The administrator has the option to enable this feature.

The password expiry alert feature provides customizable syslog log level notifications based on the configured 'max-password-age' setting, allowing for tailored alerts as passwords approach expiration.

You can configure the user password expiry alert using the **expiry-alert** command.

```
device(config-system-aaa-authentication-password-attributes)# expiry-alert
device(config-system-aaa-authentication-password-attributes-expiry)#
  critical  Critical severity log expiry alert period
  end       End current mode and change to enable mode
  exit      Exit current mode and down to parent mode
  info      Info severity log expiry alert period
  list      List all supported configuration commands
  major     Major severity log expiry alert period
  minor     Minor severity log expiry alert period
  no        Negate a command or set its defaults
  pwd       Display current mode
```

For information on password expiry alert configurations commands and syntax, see *Extreme ONE OS Switching Command Reference Guide*.

Password Configuration: Special Characters

You can configure passwords using any possible characters, consistent with Linux system standards. However:

- CLI Terminal Limitation: When entering passwords through the CLI terminal, the use of '|' and '?' characters is not supported.
- gNMI Exception: This restriction does not apply when configuring passwords through gNMI.

Configure an account to disable automatically upon Inactivity Expiry

When creating or editing an account, you can specify when the account automatically disables after it is not used (active) for a configured period of time.

There might be instances when you would like to automatically disable an account when the account is inactive for some set period of time. Inactivity means that the account has not been used, in the recent past, to access this device. Use the **acct-**

inactivity-expiry-period parameter to configure the number of days after which the account is automatically disabled (expires).

> Note
>
> The *admin* accounts cannot be disabled.
>
> - For default admin users, 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, can have either the Admin or User role.

1. In privileged EXEC mode, enter the **configure terminal** command.

   ```
   device # configure terminal
   ```

2. 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 *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. Expiry RASLOG is generated when time crosses the acct inactivity expiry period.

Configure an account with 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 will be generated when a specific user account is inactive for 20 days.

> Note
>
> Without configuring expiry period, warning period cannot be configured.

1. In the previleged EXEC mode, enter the **configure terminal** command.

   ```
   device # configure terminal
   ```

2. Enter the **user** command with the **inactivity-warning-period** command with 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. Warning RASLog is generated when time crosses the account inactivity warning period.

Changing default password for the system default accounts

The default system username is 'admin'. Upon first login, you'll be prompted to change the default password and username.

Note

Password Requirements

- **Default Minimum Length**: 8 characters
- **Complexity Requirements**:
  - Combination of alphanumeric and special characters
  - Must include both uppercase and lowercase letters

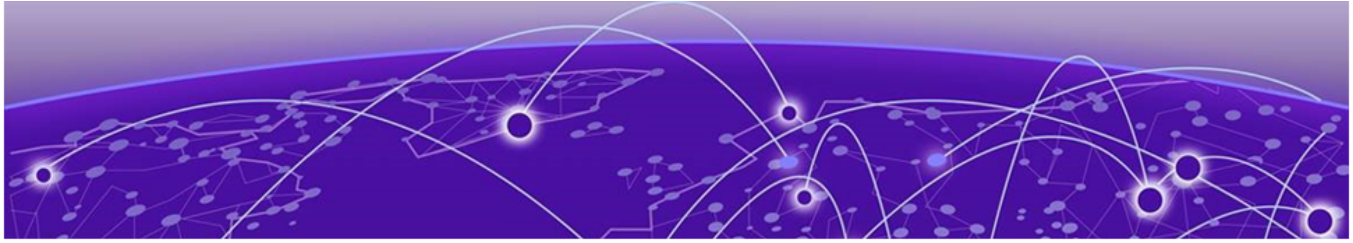
Follow this is procedure to change the default username 'admin' or update the password later.

1. To update the default password, run the following command:

   ```
   device(config-system-aaa-authentication)# admin-user admin password <password>
   ```

2. To update the default username, run the following command:

   ```
    device(config-system-aaa-authentication)# admin-user test password <password>
   ```



Northbound Interfaces & Security

Use this topic to learn about the Northbound interface components.

Northbound Interfaces

SSH

- Secure Communication: SSH enables secure communication with Extreme ONE OS, running OpenSSH_9.6p1 server. Only the northbound interface is enabled by default in the management VRF.
- Default Configuration: SSH server is enabled by default in the management VRF and is the only northbound interface enabled by default.
- User Authentication: Password-based authentication with local or remote user information storage (TACACS, LDAP, RADIUS).

Configurable SSH Parameters

- Host Key: Manually generate SSH host-key pairs, replacing existing keys. Default: RSA 4096 bits. SSH host keys are preserved in the `/etc/ssh` folder.
- Host Key Preservation: SSH host keys are preserved across firmware upgrades and full installations.
- Supported key types and sizes:
  - RSA: 2048 and 4096 bits (default)
  - ECDSA: 256 bits
- Ciphers, Key Exchange Algorithms, and Message Authentication Codes: Configurable with priority lists.
- Client Alive Interval: Terminate client connection if not reachable within the specified interval (default: 0, disabled).

SSH Server Instances

- Default Instance: Created and associated with the management VRF by default.
- Multiple Instances: Can be configured with unique VRF instances.
- Modification and Deletion: Applicable only to new SSH connections

Ciphers in SSH

Ciphers ensure data privacy over SSH connections. The default ciphers are:

- aes256-gcm@openssh.com
- aes256-ctr
- chacha20-poly1305@openssh.com
- aes192-ctr
- aes128-gcm@openssh.com
- aes128-ctr

Message Authentication Codes (MACs)

MACs ensure the integrity of messages sent over SSH connections. The default MACs are:

- hmac-sha2-512-etm@openssh.com
- hmac-sha2-256-etm@openssh.com
- hmac-sha2-512
- hmac-sha2-256

Key Exchange Algorithm

Key exchange algorithms securely exchange a shared session key with a peer. The default algorithms are:

- curve25519-sha256
- curve25519-sha256@libssh.org
- diffie-hellman-group-exchange-sha256
- diffie-hellman-group16-sha512
- diffie-hellman-group18-sha512
- diffie-hellman-group14-sha256

Other Northbound Interfaces

- Telnet: not configured by default, unencrypted, default port is 23. SSH and telnet combined session limit is 32.

  Note
  Ongoing Telnet sessions will be terminated in the following cases:

  1. Update user role for local users
  2. Delete local users
  3. Terminal timeout

- GRPC Server: not configured by default, runs over secure TLS connection

  To enable a GRPC server, a certificate-id must be associated. Additionally, Mutual TLS is disabled by default.
- SNMP: not configured by default, supports V1, V2, and V3
- TLS: supports versions 1.3 and 1.2, uses OpenSSL 3.2.3 package for traffic encryption

Note
- Configuration changes apply only to new connections, existing connections remain intact.
- Multiple SSH and Telnet server instances can be configured with unique VRF instances.

Configuration File Management

- Backup and Restore: supports backing up, viewing, and restoring configuration files
- Copy Commands:
  - Copy the running configurations to file for backup
  - Copy the configuration file to the running configurations to restore
- Display Configuration: View configurations in a file

Start Shell

- UNIX-Level Shell: capture the cli, **`start-shell`**
- Return to CLI: type "exit" to return to CLI

Service Upgrade and Restart

- Security Features: most security features will not be impacted by service restart
- Impacted Features:
  - GNMI set queries may not work during process restart
  - Certificate Manager: impacts certificate expiry alert logs
  - Local AAA: GNMI clients will not work

    - Remote AAA: new clients cannot authenticate with remote servers during restart
    - GNMI/GRPC: connections will be impacted, but GRPC certificate ID and mTLS will not be impacted if configured already

Feature-Specific Impacts

- SSH: no impact to existing clients, new clients can authenticate with old configuration
- Telnet: no impact to existing clients, new clients can authenticate once enabled before restart
- Certificate Manager: no impact to existing certificate usage, but impacts certificate expiry alert logs
- Keychain: no impact, as all configurations are in DB
- Banner/RBAC/NTP: no impact

Banner Configuration

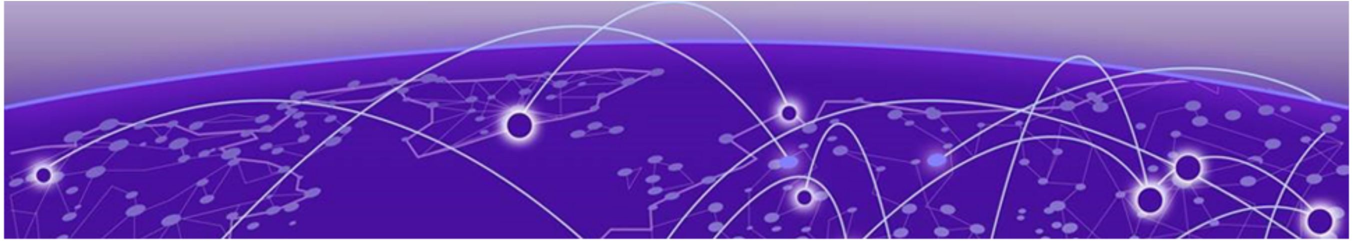
Banner configuration allows displaying text before or after connecting to the switch.

Types of Banners

1. Login Banner: displays text before login
2. Message of the Day (MOTD) Banner: displays text after successful login authentication

Password Recovery

- Local User Password Recovery: forgotten passwords for local users can be reset by logging in as the last-resort user (admin-user) and updating the password
- Admin-User Password Recovery: if the admin-user password is forgotten, the only option is to perform a net install



AAA (Authentication, Authorization, and Accounting)

Use this topic to learn about authentication, authorization, and accounting.

Authentication

Use this topic to learn about external authentication, common features, and authentication order.

External Authentication

1. TACACS+: TACACS+ is an authentication protocol that provides authentication services for users configured in remote servers.
    - Default Settings
        - Port: 49
        - Retry: 2
        - Timeout: 3 seconds
        - Maximum servers: 6
        - Default role: "user"
    - Configuration Requirements
        - Shared key should match with TACACS+ server configuration file
        - Role should be assigned to user configured on TACACS+ server (either "admin" or "user")
2. LDAP: LDAP is an open protocol used for directory services authentication.
    - Default Settings
        - Port: 389 (LDAP), 636 (secure LDAP)
        - Retry: 2
        - Timeout: 3 seconds
        - Maximum servers: 6
        - Default role: "user"

- Configuration Requirements
  - LDAP server's CA certificates should be imported in Extreme ONE OS
  - Role mapping should be performed to map LDAP user roles to available roles in Gen4OS

3. RADIUS: RADIUS is a networking protocol that authenticates remote management users.
   - Default Settings
     - Port: 1812 (RADIUS over UDP), 2083 (RADIUS over TLS)
     - Retry: 2
     - Timeout: 3 seconds
     - Maximum servers: 6
     - Default role: "user"
   - Configuration Requirements
     - RADIUS server's CA certificate should be imported in Extreme ONE OS
     - Role should be assigned to user configured on RADIUS server (either "admin" or "user")

Common Features

- In-band Support: each external authentication server can be configured with a VRF and source interface
- Failover: failover from one server to another server happens when a server fails to respond or is unreachable

Authentication Order

Authentication mode defines the order of authentication sources for user authentication.

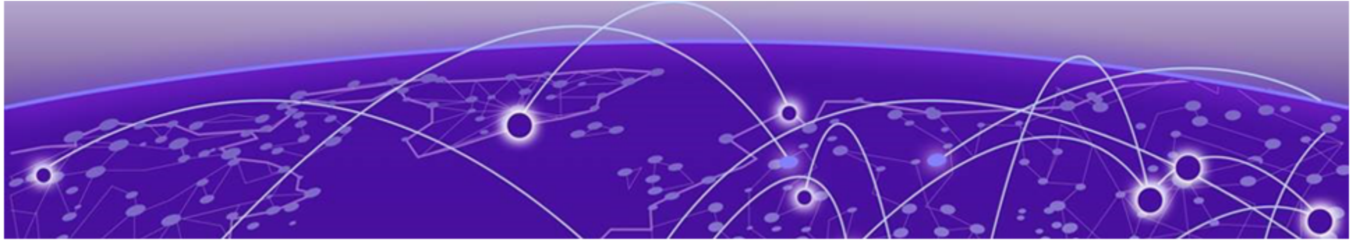
- Default Mode: local users only
- Configurable Modes: TACACS+, LDAP, RADIUS with local user as fallback
- Applicability: SSH, Telnet, and GNMI

Authorization

- Role-Based Access Control (RBAC): restricts access to resources based on assigned roles
- Predefined Roles: 2 roles - admin (read-write access) and user (read-only access)
- Role Assignment: role must be specified when creating user accounts

Accounting

- Local Accounting: all device operations are locally logged and can be viewed using "show logging" command
- Remote Accounting: TACACS+ and RADIUS accounting is supported
- Command Accounting: commands executed on the device can be tracked by enabling command accounting on TACACS+ server
- Accounting Packet Attributes: cmd (command as string) and status (status of execution)
- GNMI Limitation: GNMI activities are not logged to RADIUS server even if accounting is enabled



Routing Policy

Use this topic to learn about the routing policy.

Routing Policy Overview

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

- Routing Policy Server
  - Part of 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 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:

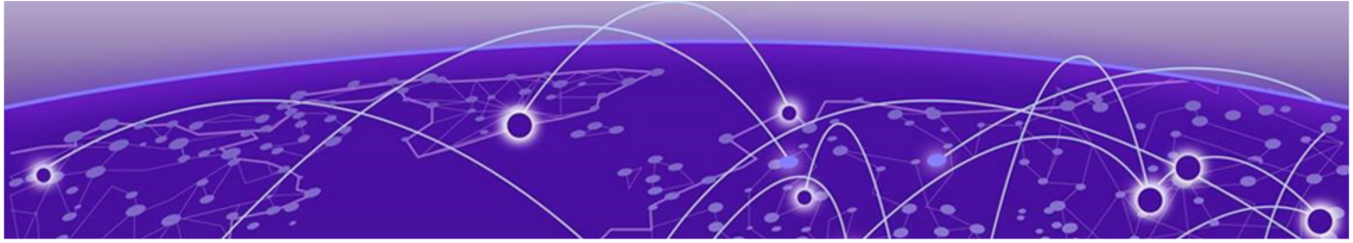
- Before placement in the routing table
- After placement in the routing table

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:

- Apply routing filters
- Evaluate routes against routing policies
- Augment/change advertised or accepted route information

> **Note**
> For details on syntax and parameters, see *Extreme ONE OS Switching v22.2.0.0 Command Reference Guide.*.

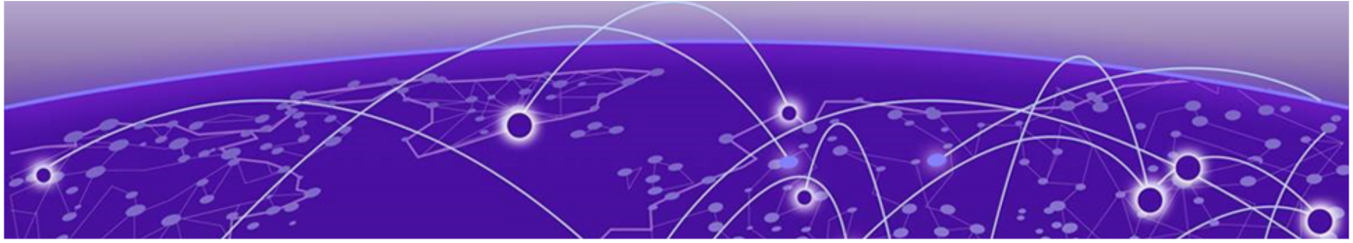


Key Chain Management

Key Chain Management Overview

Keychain management allows users to create and maintain sequences of keys for secure communication with peers.

- Configurability:
    - 128 keychains can be configured
    - Each keychain can hold up to 8 keys
    - Configurable tolerance for key authentication
- Key Rollover: keychain management provides a secure mechanism to handle key rollover based on the send and receive lifetimes of keys
- Purpose: maintain stable communications and secure data plane and control plane packets



Network Time Protocol and Clock Management

Use this topic to learn about the Network Time Protocol (NTP) and Clock management.

Clock Management

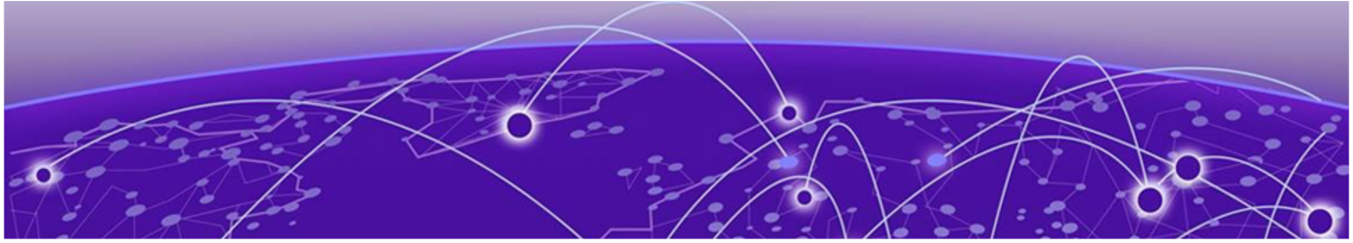
- System Timezone: can be configured to the appropriate timezone value.
- Clock Set Command: Use the clock set command if no other time sources are available.

Network Time Protocol Management

Network Time Protocol (NTP) is a protocol that synchronizes computer clock times in a network.

NTP Features

- Uses Coordinated Universal Time (UTC) to synchronize CPU clock time
- Avoids vulnerabilities in information exchange communication
- Can configure up to 8 servers and 8 peers to synchronize system clock
- Supports in-band configuration with VRF mapping to NTP service



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 allows 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.

Note
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 1.1.1.1
   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 1.1.1.1 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 1.1.1.1 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 ONE OS switching and Routing,
   CN=extremenetworks.com
   issuer=C=US, ST=CA, O=Extreme Networks, OU=Extreme ONE OS 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 ONE OS switching and Routing,
   CN=extremenetworks.com
   issuer=C=US, ST=CA, O=Extreme Networks, OU=Extreme ONE OS 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 ONE OS switching and Routing,
   CN=extremenetworks.com
   issuer=C=US, ST=CA, O=Extreme Networks, OU=Extreme ONE OS 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
   1.1.1.1 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
   1.1.1.1 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 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 ONE OS switching and Routing,O=Extreme
Networks,ST=CA,C=US
issuer=CN=extremenetworks.com,OU=Extreme ONE OS 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 ONE OS switching and Routing,O=Extreme
Networks,ST=CA,C=US
issuer=CN=extremenetworks.com,OU=Extreme ONE OS 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 ONE OS switching and Routing,O=Extreme
Networks,ST=CA,C=US
issuer=CN=extremenetworks.com,OU=Extreme ONE OS 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 1.1.1.1 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. Each SSL profile can be associated with multiple applications, allowing efficient certificate management 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:

- 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 allow LDAP client instance association.
- RADIUS: Yang data model is augmented to include an SSL profile to allow 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 allow 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 1.1.1.1
   device(config-system-aaa-server-group-ldap-server-1.1.1.1)# 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
1.1.1.1 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 1.1.1.1
        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 1.1.1.1
device(config-system-aaa-server-group-radius-server-1.1.1.1)# ssl-profile-id sp1
device(config-system-aaa-server-group-radius-server-1.1.1.1)# 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
1.1.1.1 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 1.1.1.1
        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 1.1.1.1
device(config-system-logging-remote-server-1.1.1.1)# secure-forwarding tls
ddevice(config-system-logging-remote-server-1.1.1.1)# 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
1.1.1.1 certificate /tmp/cert.crt user vikhanna password **** vrf mgmt-vrf

device# show running-config system logging remote-server
system
  logging
    remote-server 1.1.1.1
      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

   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 1.1.1.1 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 config 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 config and security audit logs with the extracted username.

Config Audit Log

```
2025-04-13 13:54:07.022 UTC +0000 LogID:6001 info Msg: xco-user/10.x.x.x/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 1.1.1.1, server-group radius 1.1.1.1, logging remote-server
   1.1.1.1"]
   ```

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 50
   - Configure Certificate Expiry Alert on page 52

Upgrading and Downgrading Certificate Management

Upgrade from 22.1.6.0 to a higher release requires gNOI to gNSI migration for all the certificates that are used by the device. Downgrade to 22.1.6.0 from a higher release also requires manual migration steps from gNSI to gNOI.

Use this topic to learn about the following procedure:

1. Upgrade and Migrate from 22.1.6.0 to a Higher Release on page 48
2. Downgrade and Migrate to 22.1.6.0 from a Higher Release on page 48

Upgrade and Migrate from 22.1.6.0 to a Higher Release

When upgrading from 22.1.6.0 to a higher release, a manual migration from gNOI to gNSI is required for all certificates used by the device. This includes:

- gRPC server application certificate
- CA certificates for gRPC mutual authentication, LDAP, RADIUS, and Syslog

1. Remove Existing Certificates: After upgrading, remove all certificates imported on the device using the command:

   ```
   certificate-manager delete all
   ```

2. Migrate Certificates for each application.

   For gRPC Server instance

   a. Import the application certificate under an SSL profile. For mutual authentication, import the CA certificate to the same profile.
   b. Associate the profile with the gRPC server instance.
   c. Restart the gRPC server instance.
   d. Alternatively, generate a certificate under the `ssl-reserved-generated` SSL profile using:

      ```
      certificate-manager generate ssl-profile-id ssl-reserved-generated certificate-
      extension san 1.1.1.1
      ```

   e. Associate the `ssl-reserved-generated` profile to the gRPC server instance and restart it.

      - LDAP and RADIUS Instances:
        - Import the CA certificate under an SSL profile.
        - Associate the profile to the LDAP/RADIUS client instance.
      - Syslog Instance:
        - Import the CA certificate under an SSL profile.
        - Associate the profile to the Syslog client instance.
        - Restart the Syslog client instance.

Downgrade and Migrate to 22.1.6.0 from a Higher Release

When downgrading to 22.1.6.0 from a higher release, a manual migration from gNSI to gNOI is required for all certificates used by the device. This includes:

- gRPC server application certificate
- CA certificates for gRPC mutual authentication, LDAP, RADIUS, and Syslog

1. Remove Existing Certificates: After downgrading, remove all certificates imported on the device using the command:

   ```
   certificate-manager delete all
   ```

2. Migrate Certificates for each application.

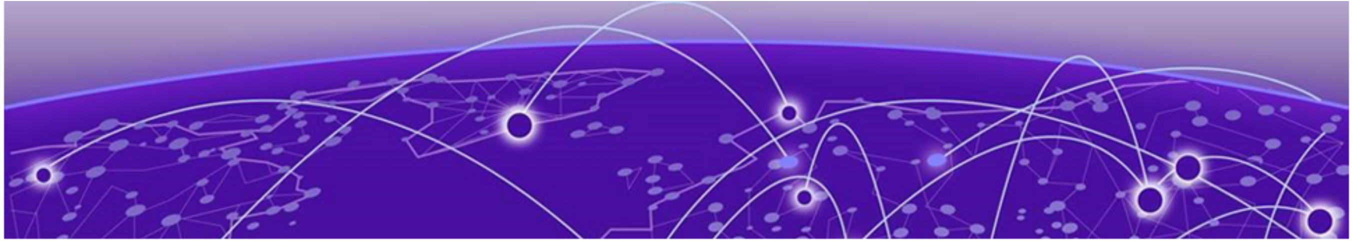
   For gRPC Server instance

   a. Import the application certificate under certificate-id.
   b. For mutual authentication, import the corresponding CA certificate to the device.
   c. Associate the certificate-id to the gRPC server instance.
   d. Restart the gRPC server instance.
   e. Alternatively, generate a certificate under the default SSL profile using:

      ```
      certificate-manager generate certificate-id default certificate-extension san
      1.1.1.1
      ```

   f. Associate the default `certificate-id` profile to the gRPC server instance and restart it.

      - LDAP and RADIUS Instances: Import the CA certificate to the device.
      - Syslog Instance:
        - Import the CA certificate to the device.
        - Restart the Syslog client instance.



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.

Note
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 may be delayed by upto 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 ONE OS does not allow importing an already expired certificate.
- If the system time is manually changed after a notification is sent, Extreme ONE OS 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 following certificates are monitored for expiry:

- app certificate
- ca-certificate

Configure Certificate Expiry Alert

Certificate expiry alerts can be configured for four (4) different 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 above configuration example:

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

Note

- 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.