



ExtremeCloud™ Orchestrator v4.1.0 Security Configuration Guide

Authentication, Authorization, and Hardening

9041113-00 Rev AA
September 2026



Copyright © 2026 Extreme Networks, Inc. All rights reserved.

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/>

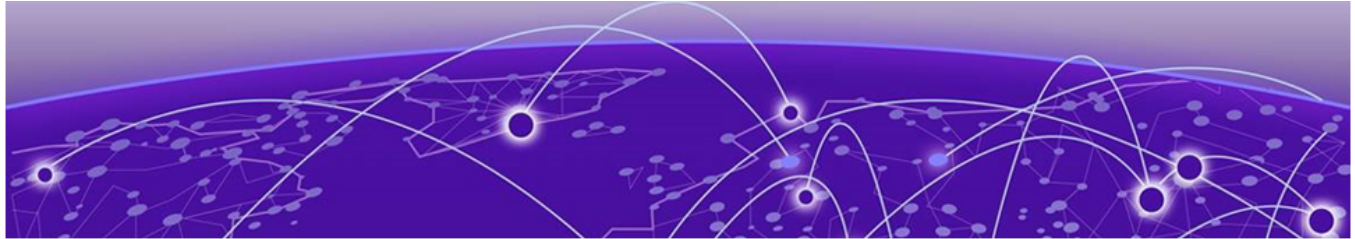
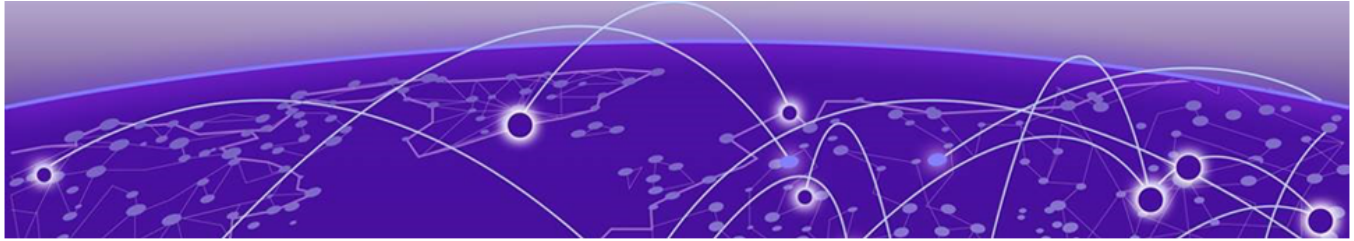


Table of Contents

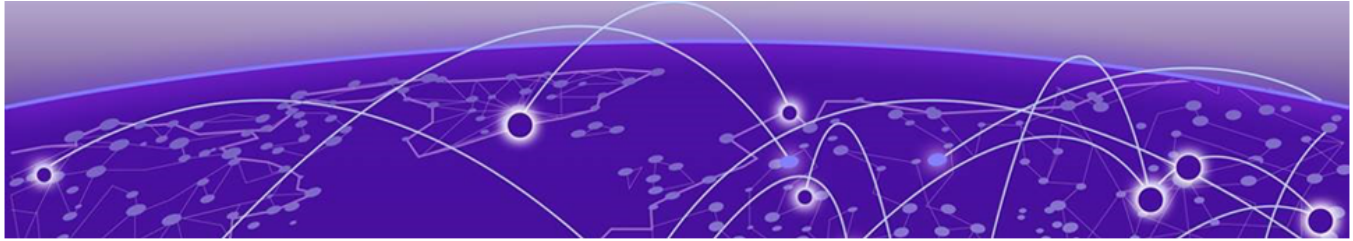
Abstract.....	v
Preface.....	vi
Text Conventions.....	vi
Documentation and Training.....	vii
Open Source Declarations.....	viii
Training.....	viii
Help and Support.....	viii
Subscribe to Product Announcements.....	ix
Send Feedback.....	ix
What's New in this Document.....	10
Security Features in XCO.....	11
XCO User Authentication and Authorization.....	11
Authentication.....	12
Authentication Policy CLI configuration.....	12
Authorization.....	14
Assign and View XCO Roles.....	14
XCO RBAC Policy Enforcement.....	15
Update TPVM Password in High Availability Deployment.....	19
Configure an External LDAP Server.....	20
TACACS+ Settings.....	25
Configure TACACS using CLI.....	25
Configure Local User using CLI.....	26
BGP MD5 Authentication.....	28
BGP MD5 Authentication on Fabric Links.....	29
BGP MD5 Authentication on Edge Links.....	38
XCO Certificate Management.....	48
Device Certificates.....	49
XCO Certificates.....	55
External Certificates.....	67
Alarms for Auto Certificate Renewal Failure.....	67
Certificate Management Using CMPv2.....	69
Security Hardening.....	78
Overview of System Hardening in XCO.....	78
System Hardening for CIS-CAT Assessments.....	79
Hardening Script for Ubuntu Linux Host Servers.....	84
Hardening Script for Ubuntu Linux Host Servers.....	84
Security Hardening for SLX-OS in XCO.....	84
SLX-OS Device Configuration.....	84
Global Device Security Settings.....	87
Device Security Settings.....	88

Drift and Reconcile for Security Settings.....	89
The iptables Policy.....	91
Nmap Scan Output from a Remote System on the VIP.....	92
Nmap Scan Output on the Active Node on a Multi-node Setup.....	92
Secure the Grub Boot Loader.....	93
Install the Linux Audit System.....	94
Secure NTP and XCO.....	95
Secure DNS and XCO.....	96
Detect Rootkits with rkhunter.....	97
Configure Remote Syslog (RELP) Securely.....	98
Configuration Files Modification Guidelines.....	100
ca.cfg.....	100
server.cfg.....	100



Abstract

This security configuration guide for ExtremeCloud™ Orchestrator version 4.1.0 outlines comprehensive procedures for implementing robust authentication, authorization, and system hardening across managed environments. It details CLI-based configuration of authentication policies using Unix, LDAP, and TACACS+, and enforces role-based access control (RBAC) for REST APIs. The guide includes instructions for managing local users, assigning roles via LDAP groups or custom attributes, and configuring BGP MD5 authentication for both fabric and edge links, with phased deployment strategies to minimize network disruption. System hardening techniques cover Ubuntu Linux and SLX-OS devices, including CIS-CAT assessments, iptables firewall policies, secure GRUB bootloader settings, auditd logging, OSSEC HIDS, authenticated NTP, encrypted DNS via dnscrypt-proxy, and rootkit detection with rkhunter. Designed for technically proficient administrators, the document ensures secure deployment and operation of XCO in both Clos and non-Clos topologies, with emphasis on configuration idempotency, drift reconciliation, and secure password handling.



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 <i>enter</i> and <i>type</i>	When you see the word <i>enter</i> 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 <i>type</i> .
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.
<i>italic</i> 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, <i>member[member...]</i> .
\	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.

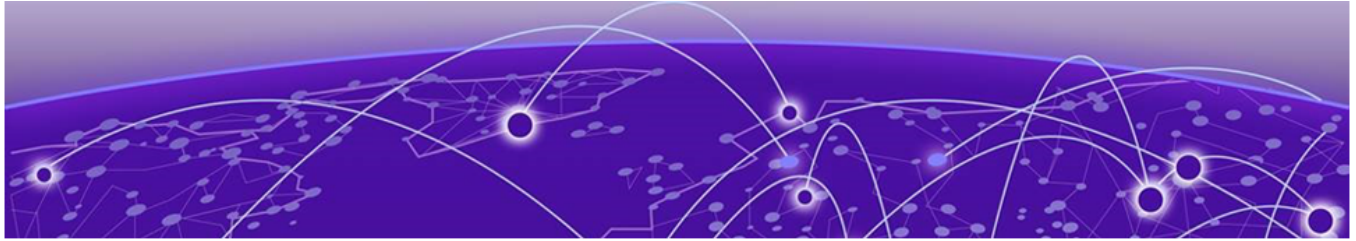
Send Feedback

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 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.



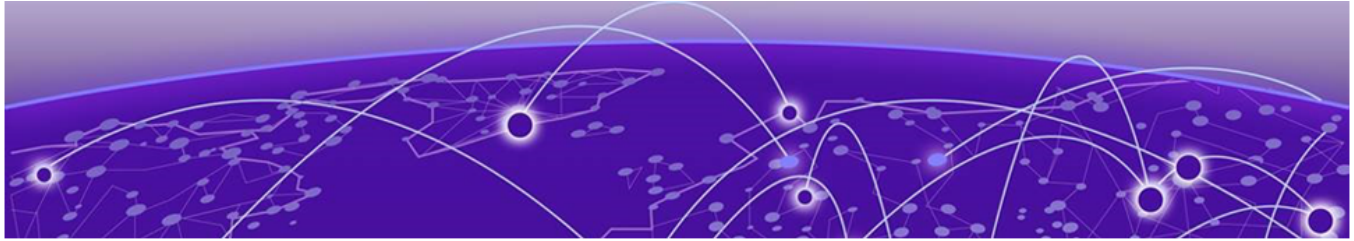
What's New in this Document

As of version 4.0.2, the ExtremeCloud Orchestrator (XCO) no longer includes support for the Visibility Skill. This deprecation follows the official End-of-Sale (EOS) announcement issued on February, 2025.

The following table describes information updated to this guide for the ExtremeCloud Orchestrator (XCO) 4.1.0 software release.

Table 4: Summary of changes

Description	Link
System Hardening for CIS-CAT Assessments	System Hardening for CIS-CAT Assessments on page 79



Security Features in XCO

[XCO User Authentication and Authorization](#) on page 11

[BGP MD5 Authentication](#) on page 28

[XCO Certificate Management](#) on page 48

The following section provides an overview of the security features in ExtremeCloud Orchestrator (XCO).



Note

From release 3.2.0 onward, Extreme Fabric Automation (EFA) is referred to as ExtremeCloud Orchestrator (XCO). The terms EFA and XCO refer to the same product and are used interchangeably.

XCO is always installed in secure mode and is operational. For details on how to achieve this, refer to the [ExtremeCloud Orchestrator CLI Administration Guide, 4.1.0](#), which includes details on XCO security options and commands.

The following section provides detailed information on security features:

- **Authentication and authorization:** Explains how XCO users are validated and managed with Role-based Access Control (RBAC).
- **BGP MD5 authentication on edge links:** How to authenticate all the BGP peer and peer-group used for edge connectivity.
- **BGP MD5 authentication on fabric links:** How to use MD5 for BGP connections across all fabric links.

ExtremeCloud Orchestrator encrypts sensitive service credentials used for internal service-to-service authentication. For production deployments, Extreme Networks recommends supplying a deployment passphrase during installation or upgrade to provide additional protection for stored credentials. For more information, see *ExtremeCloud Orchestrator Deployment Guide*.

XCO User Authentication and Authorization

At installation time, starting with EFA 2.5.0, all XCO users of services such as MySQL and RabbitMQ are assigned random passwords that are stored in XCO configuration files. This satisfies the requirement to enforce the change of default passwords, and no two XCO installations share identical passwords.

For more information, see [XCO RBAC Policy Enforcement](#) on page 15 and [Assign and View XCO Roles](#) on page 14.

Authentication

XCO validates users and their credentials using the following mechanisms:

- Unix authentication (local and remote) on the host where XCO is installed. Host credentials are the default validation method if LDAP validation fails.
- External LDAP server: Users configured in LDAP use their LDAP credentials to log in to XCO.
- Authentication support:
 1. External authentication:
 - a. LDAP
 - b. TACACS
 - c. Auth preference and fallback

The following graphic illustrates how users can check all modes of authentication at the same time. Define the authentication preference to help users to configure multiple modes of authentication at once.

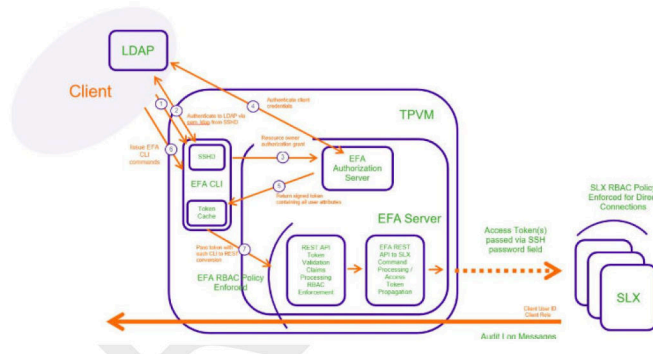


Figure 1: LDAP authentication example

Operational or maintenance tasks are propagated to SLX-OS devices through OAuth2 and JWT access tokens. TLS is used for connections with SLX-OS devices. When XCO is installed in secure mode, traffic to northbound interfaces uses TLS. For more information about secure mode, see the "XCO Installation Modes" topic in the [ExtremeCloud Orchestrator Deployment Guide, 4.1.0](#).

Authentication Policy CLI configuration

For CLI users, the auth preference configuration is available under `efa auth authentication preference`.

```
KVM:~$ efa auth authentication preference

Available Commands:
  add      Add the authentication preference
  update   Update the authentication preference
  delete   Delete the authentication preference
  show     show authentication preference
```

Show authentication preference:

```
KVM:~$ efa auth authentication preference show
+-----+-----+-----+
| Auth Type | Identifier | Preference |
```

```
+-----+-----+-----+
| HOST   | HOST   | 1      |
+-----+-----+-----+
```

Add authentication preference:

```
KVM:~$ efa auth authentication preference add --authType=LOCAL --identifier=LOCAL --
preference=3
Successfully updated the auth preference.
```

```
+-----+-----+-----+
| Auth Type | Identifier | Preference |
+-----+-----+-----+
| LOCAL     | LOCAL     | 3          |
+-----+-----+-----+
```

Show authentication preference:

```
KVM:~$ efa auth authentication preference show
```

```
+-----+-----+-----+
| Auth Type | Identifier | Preference |
+-----+-----+-----+
| HOST      | HOST      | 1          |
+-----+-----+-----+
| LOCAL     | LOCAL     | 3          |
+-----+-----+-----+
```

Update authentication preference:

```
KVM:~$ efa auth authentication preference update --authType=LOCAL --identifier=LOCAL --
preference=2
Successfully updated the auth preference.
```

```
+-----+-----+-----+
| Auth Type | Identifier | Preference |
+-----+-----+-----+
| LOCAL     | LOCAL     | 2          |
+-----+-----+-----+
```

```
KVM:~$ efa auth authentication preference delete --authType=LOCAL --identifier=LOCAL
```

Add authentication preference for TACACS authentication:

```
efa auth authentication preference add --authType=TACACS --identifier=10.37.135.12 --
preference=3
Successfully added the auth preference.
```

```
+-----+-----+-----+
| Auth Type | Identifier | Preference |
+-----+-----+-----+
| TACACS    | 10.37.135.12 | 3          |
+-----+-----+-----+
```

```
efa auth authentication preference add --authType=LDAP --identifier=kvm12.com --
preference=4
Successfully added the auth preference.
```

Add authentication preference for LDAP authentication:

```
efa auth authentication preference add --authType=LDAP --identifier=kvm12.com --
preference=4
Successfully added the auth preference.
```

```
+-----+-----+-----+
```

```

| Auth Type | Identifier | Preference |
+-----+-----+-----+
| LDAP      | kvm12.com | 4          |
+-----+-----+-----+

```

Show authentication preference:

```

efa auth authentication preference show
+-----+-----+-----+
| Auth Type | Identifier | Preference |
+-----+-----+-----+
| HOST      | HOST      | 1          |
+-----+-----+-----+
| LOCAL     | LOCAL     | 2          |
+-----+-----+-----+
| TACACS    | 10.37.135.12 | 3          |
+-----+-----+-----+
| LDAP      | kvm12.com | 4          |
+-----+-----+-----+

```

Authorization

After XCO is deployed, the installer has the role of SystemAdmin and has complete access to XCO functionality. For installation on TPVM, this user has the username 'extreme'. By default, no other host OS users can access XCO unless the SystemAdmin assigns the appropriate roles. RBAC occurs on XCO and API.

Assign and View XCO Roles

You can assign a role to a user and to an LDAP group. For more information about XCO roles, see [XCO RBAC Policy Enforcement](#) on page 15.

1. To assign a role to a user, run the following command:

```

# efa auth rolemapping add --name=user2 --role SystemAdmin -auth-type HOST
Successfully added the role mapping

```

In this example, a user named user2 was assigned the role of SystemAdmin.

2. To assign a role to an LDAP group, run the following command:

```

# efa auth rolemapping add --name "cn=viewer,dc=extr,dc=com" --role NetworkOperator
--type group
--auth-type LDAP --auth-identifier ldapconfig
Successfully added the role mapping.

```

In this example, a group named "cn=viewer,dc=extr,dc=com" was assigned the role of NetworkOperator.

3. To view all role assignments, run the following command:

```

# efa auth rolemapping show
+-----+-----+-----+-----+-----+-----+
| ID | Name          | Role           | Type | Auth Type | Auth Identifier |
+-----+-----+-----+-----+-----+-----+
| 1 | efauser      | SystemAdmin    | USER | HOST      |                 |
+-----+-----+-----+-----+-----+-----+
| 2 | fabricuser   | FabricAdmin    | USER | LOCAL     |                 |
+-----+-----+-----+-----+-----+-----+
| 3 | viewer       | NetworkOperator | GROUP | TACACS    | 10.x.x.x        |
+-----+-----+-----+-----+-----+-----+
| 5 | cn=viewer,dc=extr,dc=com | NetworkOperator | GROUP | LDAP      | ldapconfig      |

```

```
| +---+-----+-----+-----+-----+-----+
```

4. To delete a role assignment, run the following command:

```
# efa auth rolemapping remove --id 3
Deleted role mapping successfully
```

In this example, the role for the user with ID 3 was removed.

XCO RBAC Policy Enforcement

XCO implements an RBAC policy governing access to northbound REST APIs.

The RBAC policy is enforced at the northbound interface, immediately after validation of the access token. An error message is returned if the RBAC permission check fails.

Security Troubleshooting

Use the following logs to troubleshoot authentication, authorization, or RBAC issues.

Table 5: Security log locations

Log source	Filepath
XCO server	/var/log/efa/auth/auth-server.log /var/log/efa/rbac/rbac-server.log
XCO TPVM	/apps/efa_logs/auth/auth-server.log /apps/efa_logs/rbac/rbac-server.log
SLX-OS device	/var/log/pam-oauth2.log

Use the following commands to see the list of commands that were run during a specified time and identify potential causes for issues, such as when an RBAC error occurred.

- **efa auth execution show**
- **efa rbac execution show**
- **efa inventory execution show**

RBAC and REST URI Matrix

The RBAC policy is expressed in a permissions matrix indexed by RBAC role and REST URI, in which each matrix element enumerates the permitted HTTP methods.

Table 6: RBAC and REST Matrix

	Role A	Role B	Role C
REST URI 1	GET	GET	GET, POST, PUT, PATCH, DELETE
REST URI 2	GET, POST	GET, POST, PUT	GET, POST, PUT, PATCH, DELETE
REST URI 3	GET, POST	GET, POST	GET, POST, PUT, PATCH, DELETE

RBAC Roles

Roles can be populated into the upstream LDAP instance.

**Note**

The SystemAdmin and NetworkOperator roles are applicable for VM mode of installation.

Table 7: Role definitions

Role	Description
FabricAdmin	- Registers devices to the fabric - Configures fabric parameters - Validates all devices in the fabric - Configures switches for IP fabric with overlay and without overlay - Creates tenants - Creates networks inside tenants, such as VRF, EPG, and PO - Performs fabric debug activities - Has privileges for Hyper-V and vCenter operations
SecurityAdmin	Performs user management, PKI, and key management operations
NetworkOperator	- Has view-only privileges for fabric configurations, information for tenants and inventory, and all ecosystem information - Cannot make changes in the system
SystemDebugger	- Has privileges to perform supportsave and system backup, and to view the running system configurations - Has privileges to perform fabric debug operations - Sets debug levels for services - Has privileges to collect execution logs from services
SystemAdmin	Has complete privileges to all operations in the system
<Tenant>Admin * Created dynamically per tenant	Performs tenant administration within the assigned tenant, such as the following: - Adds networks to the tenant - Configures network parameters - Configure switches with tenant-specific information Cannot perform actions for any other tenant

* Tenant Administrator roles are added dynamically to the system when a tenant is created. The name of the role is presented in the <Tenant-name>Admin format. For example, if a tenant with the name "RegionOne" is created, the role created for the Tenant Administrator is "RegionOneAdmin".

**Note**

You cannot create custom roles.

*Role Permissions***Table 8: Role permissions for fabric manager**

Allowed Privileges	System Admin	Fabric Admin	Tenant Admin	Network Operator	Security Admin	System Debugger
Create, clone, delete fabric in the system	✓	✓				
Register, unregister devices in fabric, configure IP fabric on the device	✓	✓				
Add, delete, and update location	✓	✓				
Show IP fabric physical, underlay, overlay topology, IP fabric configs and devices in IP fabric	✓	✓		✓		
Debug fabric operations	✓	✓				✓
Inventory, asset service operations	✓	✓				
Run CLI access on the device	✓	✓				
Create, delete, update tenants	✓	✓				✓
Create, delete EPG, PO, VRFs inside tenant	✓	✓	✓			
Add, remove port, port channels to and from EPG	✓	✓	✓			
Add, remove network policies to EPG	✓	✓	✓			
Detach network from EPG	✓	✓	✓			
Identify drift in device configuration	✓	✓				
Set tenant debug level	✓	✓	✓			✓
Create, delete router interfaces	✓	✓	✓			
View vCenter details, events, ESXI details, physical links, virtual links, disconnected links, get server settings	✓	✓	✓	✓		

Table 8: Role permissions for fabric manager (continued)

Allowed Privileges	System Admin	Fabric Admin	Tenant Admin	Network Operator	Security Admin	System Debugger
Register, delete, update vCenter	✓	✓	✓			
Set vCenter debug level	✓	✓	✓			✓
Update vCenter polling frequency, dead link clearing time	✓	✓	✓			
View SCVMM server details, service settings, physical links, virtual links	✓	✓	✓	✓		
Register, delete, update SCVMM server	✓	✓	✓			
Update SCVMM server polling frequency	✓	✓	✓			
User management, assign roles to users, configure LDAP, configure TACACS+, view available roles in the system	✓			✓	✓	
Notification service (add, delete subscribers)	✓	✓				
Execution log view	✓	✓ (No Auth and RBAC)	✓ (only Tenant)	✓	✓ (only Auth and RBAC)	✓
Support save collection	✓	✓	✓	✓	✓	✓
Backup and restore operation	✓	✓ (only backup)				✓
Install certificates	✓	✓			✓	

Update TPVM Password in High Availability Deployment

Update the TPVM password at both the Linux and SLX-OS configuration levels to ensure credential synchronization and upgrade persistence.

**Caution**

Changing only the Linux password does not update the SLX-OS configuration. This may result in password rollback during configuration replay or upgrade.

- Ensure console access to both SLX-OS switches hosting the TPVM instances.
- Plan a maintenance window. Removing trusted-peer temporarily causes the standby EFA node to go down.

This procedure applies to HA deployments where two TPVM instances operate in active/standby mode.

For details on changing TPVM password after a TPVM upgrade, see "TPVM Complete Package Upgrade" and "Change TPVM Password from XCO" in the ExtremeCloud Orchestrator Deployment Guide, 4.0.2.

1. Verify the HA state.

```
efa status
```

2. Remove trusted-peer configuration from both switches.

```
configure terminal
tpvm
no trusted-peer
```

The standby EFA node transitions to down state after this command.

3. On the switch hosting the standby TPVM instance, update the TPVM password.

```
configure terminal
tpvm
password NEWPASSWORD
```

4. For SLX-OS versions earlier than 20.6.3, restart the TPVM instance.

```
tpvm stop
tpvm start
```

From SLX-OS 20.6.3 onward, TPVM does not need to be stopped before updating the password.

5. Repeat the password update on the switch hosting the active TPVM instance.
6. Reconfigure trusted-peer with the new password.

```
trusted-peer ip <peer-ip> password NEWPASSWORD
```

7. Verify that both TPVM instances are operational.

```
efa status
```

The TPVM password is synchronized at both Linux and SLX-OS configuration levels, and HA operation is restored.

Configure an External LDAP Server

You can configure an LDAP server for user validation and to fetch user groups.

LDAP supports three modes for fetching the roles assigned to a user.

- The role is available as an attribute in the user Distinguished Name (DN) entry. Group attribute definition is not needed.
- The user has a "memberOf" attribute or any appropriate group DN attribute to identify the groups assigned to the user. Assign the corresponding LDAP group to a role in XCO.
- LDAP groups have user entries in their group definitions. Assign the LDAP groups to roles in XCO.



Note

If you configure LDAP server over SSL, and use IP to connect to the server, ensure that the certificate includes the IP as part of the subject alternative names (SANs) for a successful connection.

For more information about commands and supported parameters, see [ExtremeCloud Orchestrator Command Reference, 4.1.0](#).

Basic Configuration

Attribute	Description	Default Value
name	Unique identifier for LDAP configuration in XCO.	–
host	IPv4 or IPv6 Address/Hostname of the LDAP server.	–
port	Port at which the LDAP server is listening for connections.	389
timeout	Duration in number of seconds before considering the server unreachable.	5
bind-user-name	Distinguished Name (DN) of the user that should be used to bind, search, and retrieve LDAP entries.	–
bind-user-password	Password of the bind user.	

TLS Configuration

Attribute	Description	Default Value
tls	Use LDAP over SSL/TLS.	–
cacert	Local path to the CA certificate file for SSL verification.	–
insecuretls	Option to skip certificate validation while connecting to the LDAP server	false

Authentication

Attribute	Description	Default Value
user-search-base	Distinguished Name of the node in your directory tree from which to start searching for user objects.	-
user-object-base	Name of the object class used for user objects.	inetOrgPerson
user-login-attribute	Attribute whose value matches the username part of credentials entered by your users when logging in.	uid

Examples

To enable LDAP for authentication in XCO with OpenLDAP, use the following command:

```
efa auth ldapconfig add --name ldap_xco --host 10.x.x.x --bind-user-name
cn=ldapuser,dc=xxx,dc=com
--bind-user-password ***** --user-search-base ou=people,dc=xxx,dc=com
```

To enable LDAP for authentication in XCO with Windows AD, use the following command:

```
efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --bind-user-name
CN=ldapuser,CN=Users,DC=xxx,DC=com
--bind-user-password ***** --user-search-base CN=Users,DC=xxx,DC=com --user-object-
class user
--user-login-attribute sAMAccountName
```

To use the same configuration with TLS enabled:

```
efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --tls --cacert root-ca.pem --
bind-user-name
CN=ldapuser,CN=Users,DC=xxx,DC=com --bind-user-password ***** --user-search-base
CN=Users,DC=xxx,DC=com
--user-object-class user --user-login-attribute sAMAccountName
```

To skip certificate verification over the encrypted connection, use the following command:

```
efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --tls --cacert root-ca.pem --
insecuretls
--bind-user-name CN=ldapuser,CN=Users,DC=xxx,DC=com --bind-user-password *****
--user-search-base CN=Users,DC=xxx,DC=com --user-object-class user --user-login-attribute
sAMAccountName
```

Examples

To enable LDAP for authentication in XCO with OpenLDAP, use the following command:

```
efa auth ldapconfig add --name ldap_xco --host 10.x.x.x --bind-user-name
cn=ldapuser,dc=xxx,dc=com
--bind-user-password ***** --user-search-base ou=people,dc=xxx,dc=com
```

To enable LDAP for authentication in XCO with Windows AD, use the following command:

```
efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --bind-user-name
CN=ldapuser,CN=Users,DC=xxx,DC=com
--bind-user-password ***** --user-search-base CN=Users,DC=xxx,DC=com --user-object-
```

```
class user
--user-login-attribute sAMAccountName
```

To use the same configuration with TLS enabled:

```
efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --tls --cacert root-ca.pem --
bind-user-name
CN=ldapuser,CN=Users,DC=xxx,DC=com --bind-user-password ***** --user-search-base
CN=Users,DC=xxx,DC=com
--user-object-class user --user-login-attribute sAMAccountName
```

To skip certificate verification over the encrypted connection, use the following command:

```
efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --tls --cacert root-ca.pem --
insecuretls
--bind-user-name CN=ldapuser,CN=Users,DC=xxx,DC=com --bind-user-password *****
--user-search-base CN=Users,DC=xxx,DC=com --user-object-class user --user-login-attribute
sAMAccountName
```

Authorization

There are multiple ways to define authorization for authenticated users.

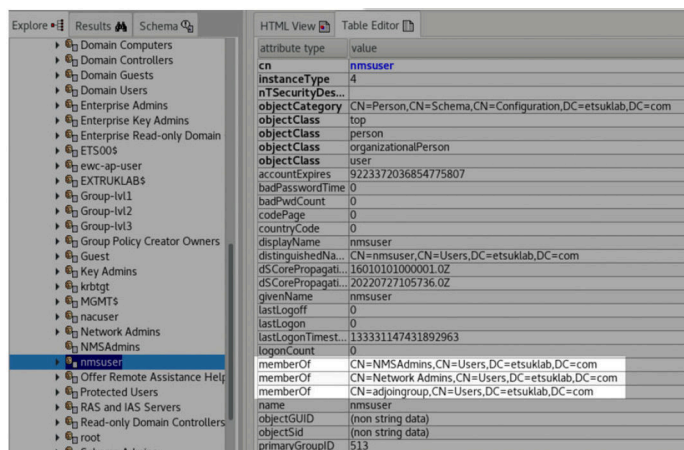
1. Assign roles to users using LDAP groups when users hold group membership details:

Attribute	Description	Default Value
user-member-attribute	Attribute to read the member of the group the user is part of.	-

Example

In Windows AD, if the user has an attribute 'memberOf' which gives the groups that he belongs to, then define 'user-member-attribute'

```
efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --bind-user-name
CN=ldapuser,CN=Users,DC=xxx,DC=com --bind-user-password *****
--user-search-base CN=Users,DC=xxx,DC=com --user-object-class user
--user-login-attribute sAMAccountName --user-member-attribute memberOf
```



These groups should be mapped to XCO roles using the role mapping command.

```
efa auth rolemapping add --name CN=NMSAdmins,CN=Users,DC=etsuklab,DC=com --role
SystemAdmin
--type group --auth-type LDAP --auth-identifier ldap_winad
```

Assign roles for multiple groups, if required.

2. To enable LDAP for authentication in XCO with OpenLDAP, use the following command:

```
efa auth ldapconfig add --name ldap_xco --host 10.x.x.x --bind-user-name
cn=ldapuser,dc=xxx,dc=com
--bind-user-password ***** --user-search-base ou=people,dc=xxx,dc=com
```

To enable LDAP for authentication in XCO with Windows AD, use the following command:

```
efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --bind-user-name
CN=ldapuser,CN=Users,DC=xxx,DC=com
--bind-user-password ***** --user-search-base CN=Users,DC=xxx,DC=com --user-object-
class user
--user-login-attribute sAMAccountName
```

To use the same configuration with TLS enabled:

```
efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --tls --cacert root-ca.pem
--bind-user-name
CN=ldapuser,CN=Users,DC=xxx,DC=com --bind-user-password ***** --user-search-base
CN=Users,DC=xxx,DC=com
--user-object-class user --user-login-attribute sAMAccountName
```

To skip certificate verification over the encrypted connection, use the following command:

```
efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --tls --cacert root-ca.pem
--insecuretls
--bind-user-name CN=ldapuser,CN=Users,DC=xxx,DC=com --bind-user-password *****
--user-search-base CN=Users,DC=xxx,DC=com --user-object-class user --user-login-
attribute sAMAccountName

efa auth ldapconfig add --name ldap_winad --host 10.x.x.x --bind-user-name
CN=ldapuser,CN=Users,DC=xxx,DC=com --bind-user-password *****
--user-search-base CN=Users,DC=xxx,DC=com --user-object-class user
--user-login-attribute sAMAccountName --user-member-attribute memberOf
```

attribute type	value
cn	nmsuser
instanceType	4
nTSecurityDes...	
objectCategory	CN=Person,CN=Schema,CN=Configuration,DC=etsuklab,DC=com
objectClass	top
objectClass	person
objectClass	organizationalPerson
objectClass	user
accountExpires	9223372036854775807
badPasswordTime	0
badPwdCount	0
codePage	0
countryCode	0
displayName	nmsuser
distinguishedNa...	CN=nmsuser,CN=Users,DC=etsuklab,DC=com
dSCorePropagati...	16010101000001.02
dSCorePropagati...	20220727105736.02
givenName	nmsuser
lastLogoff	0
lastLogon	0
lastLogonTimest...	133331147431892963
logonCount	0
memberOf	CN=NMSAdmins,CN=Users,DC=etsuklab,DC=com
memberOf	CN=Network Admins,CN=Users,DC=etsuklab,DC=com
memberOf	CN=adjoininggroup,CN=Users,DC=etsuklab,DC=com
name	nmsuser
objectGUID	(non string data)
objectSid	(non string data)
primaryGroupID	513

These groups should be mapped to XCO roles using the role mapping command.

```
efa auth rolemapping add --name CN=NMSAdmins,CN=Users,DC=etsuklab,DC=com --role
SystemAdmin
--type group --auth-type LDAP --auth-identifier ldap_winad
```

Assign roles for multiple groups, if required.

3. Assign roles to users using LDAP groups when the groups are in a different search base:

Attribute	Description	Default Value
group-search-base	Distinguished Name of the node in your directory tree from which to start searching for group objects.	-
group-object-class	object class used for group objects.	groupOfNames
group-attribute	Attribute to define search filter on group.	cn
group-member-user-attribute	Name of the user attribute whose format matches the group members.	entrydn
group-member-mapping-attribute	Name of the group attribute containing the members of a group.	member

```
efa auth ldapconfig add --name ldap_xco --host 10.x.x.x --bind-user-name
cn=ldapuser,dc=xxx,dc=com
--bind-user-password ***** --user-search-base ou=people,dc=xxx,dc=com --group-search-
base
ou=groups,dc=extrnet,dc=com
```

To override the defaults for different LDAP:

```
efa auth ldapconfig add --name ldap_xco --host 10.x.x.x --bind-user-name
cn=ldapuser,dc=xxx,dc=com
--bind-user-password ***** --user-search-base ou=people,dc=xxx,dc=com --group-search-
base
ou=groups,dc=extrnet,dc=com --group-member-user-attribute dn --group-member-mapping-
attribute memberId --group-object-class posixGroup
```

Assign the required roles for the groups in XCO using the role mapping command.

4. Assign roles to user from a custom attribute and XCO role defined with a key/value pair:

Attribute	Description	Default Value
user-role-attribute	Attribute to read the role of user from.	-
user-role-attribute-key	Attribute to read the role value from role attribute.	-

```
efa auth ldapconfig add --name ldap1 --host 10.x.x.x --bind-user-name
cn=admin,dc=extrnet,dc=com
--bind-user-password ***** --user-search-base ou=people,dc=extrnet,dc=com
--user-role-attribute role --user-role-attribute-key rolename
```

Here role is the custom schema defined in LDAP as an attribute for user and the rolename is where it holds the XCO role in LDAP.

The role attribute for the user entry in LDAP has the value of *rolename:SystemAdmin,rolename:FabricAdmin*

5. To defines roles in XCO and skip authorization in LDAP, add the required role for each user:

```
efa auth rolemapping add --name=testuser --role=FabricAdmin --type=user --auth-
type=ldap
--auth-identifier=ldap_xco
```

TACACS+ Settings

Terminal Access Controller Access-Control System Plus (TACACS+) is an external authentication server used for verifying user credentials.

The TACACS+ protocols support environments that are configured for authentication, authorization, and accounting (AAA) services.

XCO supports TACACS+ authentication in the following ways.

- XCO supports up to five auth preferences and TACACS+ servers can be added accordingly. If any TACACS+ server addition fails due to auth preference limit, delete the unwanted auth preference and add a new TACACS+ config.
- TACACS+ authentication must be enabled. If TACACS+ authentication is not enabled, only local authentication is used.
- If remote authentication fails, XCO attempts to use local authentication, which is successful only if the user is in the XCO database.
- The *secret key* configured for XCO must match the *secret key* from the TACACS+ server configuration file. Authentication fails if the two values do not match.
- The *service* and *xco-role* entries configured for XCO must match the equivalent entries in the TACAS+ server configuration file.

Configure TACACS using CLI

Only users with the role SecurityAdmin or SystemAdmin can perform this task.



Note

For details about the command and its parameters, see the [ExtremeCloud Orchestrator Command Reference, 4.1.0](#).

1. Run the following command:

```
efa auth tacacsconfig add --host 10.24.15.200 --port 49 --secret sharedsecret --
protocol CHAP
```

The command validates the attributes. If the validation is successful, the attributes are saved in the database. These details are used to validate user credentials and fetch the user role during token generation.

2. Run the following role mapping command to map TACACS server roles with the XCO roles:

```
efa auth tacacsconfig rolemapping add --host 10.24.15.200 --tacacsRole=tacAdmin --
xcoRole SystemAdmin
```

The rolemapping command validates whether or not the host is already configured in XCO. If yes, then the command maps the TACACS role with the XCO supported role. Similarly, the deletion of the host from TACACS config also deletes the TACACS roles of the host already configured using role mapping.

Example:

```
efa auth rolemapping add --name=tacAdmin --role SystemAdmin --auth-type TACACS
--auth-identifier 10.37.32.51
Successfully added the role mapping.
```

```
+-----+-----+
| attribute | value |
+-----+-----+
| id        | 5 |
+-----+-----+
| name      | tacAdmin |
+-----+-----+
| role      | SystemAdmin |
+-----+-----+
| type      | GROUP |
+-----+-----+
| auth type | TACACS |
+-----+-----+
| identifier| 10.37.32.51 |
+-----+-----+
```

```
efa auth rolemapping show
```

```
+-----+-----+-----+-----+-----+
+-----+
| ID | Name | Role |
| Type | Auth Type | Auth Identifier |
+-----+-----+-----+-----+
+-----+
| 3 | admin | SystemAdmin |
| GROUP | TACACS | 10.37.32.51 |
+-----+-----+-----+-----+
+-----+
| 5 | tacAdmin | SystemAdmin |
| GROUP | TACACS | 10.37.32.51 |
+-----+-----+-----+-----+
+-----+
| 1 | user | SystemAdmin |
| USER | HOST | |
+-----+-----+-----+-----+
```

Configure Local User using CLI

Manage local users using the cli.

Only users with the role SecurityAdmin or SystemAdmin can perform this task.

1. Add user:

For the local user creation, username, email id, and user role parameters are mandatory. The username and email id must be unique.

```
efa auth user add --userName user1 --emailID user1@test.com --roles SystemAdmin
Password: ****
Successfully added the user configuration
```


7. Reset password:

Only a user with the SystemAdmin role can reset the password of local users.

```
efa auth user reset-password --userName user1 --emailID user1@test.com
Password:
Successfully updated the user configuration
```

8. Show users:

Display the local users with details.

```
efa auth user show
+-----+-----+-----+-----+
+-----+
| Username | Roles           | Is Blocked | Email ID
| Mobile Number | Organization |
+-----+-----+-----+-----+
+-----+
| testabc  | SystemAdmin     | false      | testabc@test.com
|          |                 |            |
+-----+-----+-----+-----+
+-----+
| user1    | NetworkOperator | false      | user1@test.com
|          |                 |            |
+-----+-----+-----+-----+
+-----+
```

9. Delete user:

Only a user with the role of SystemAdmin or SecurityAdmin can delete a local user.

```
efa auth user delete --userName=user1
Deleted User configuration successfully.
```

BGP MD5 Authentication

The Border Gateway Protocol (BGP) is an exterior gateway protocol designed to exchange routing and reachability information among autonomous systems on the internet. The following table provides a list of some of the threats against BGP.



Note

BGP depends on TCP as its transport protocol. Therefore, it is vulnerable to the same security attacks as any TCP-based protocol.

Threats against BGP	Description
Denial of Service (DoS)	A malicious host sends unexpected or unwanted BGP traffic to a neighbor in an attempt to saturate control plane resources, which results in not having enough resources to process legitimate BGP traffic on the neighbor.
Route Manipulation	A malicious host modifies the contents of a BGP routing table, diverting traffic, and preventing it, without the sender's knowledge, from reaching its intended destination.

Threats against BGP	Description
Route Hijacking	A rogue BGP neighbor maliciously advertises a victim's networks to redirect some or all of victim's traffic to itself.
Misconfiguration (non-malicious)	An unintentionally misconfigured BGP router could affect the Internet's BGP routing table, possibly leading to network outages and, worse, unauthorized access to the network traffic.

BGP authentication enables the routers to share information only if they can verify that they are communicating to a trusted source, based on a password. Successful authentication between BGP neighbors proves that the neighbors are legitimate and trusted, verifies communications between those neighbors, and ensures that only routes learned from legitimate neighbors are added to the routing table. Authentication must be enabled on both sides of the peering session and the same password must be used on both peers.



Note

For BGP MD5 passwords, the ASCII characters 0-32 are not supported. In addition, special handling is required for MD5 passwords that contain certain special characters.

Examples

MD5 password provided through CLI	Actual MD5 password
'~!@#\$%^&*()_-=[]\ '<>/''''	~!@#\$%^&*()_-=[]\ '<>/'
'a'"a"	a"a
'a'"'"a"	a'"a"
'a'"'"'"a"	a'"a"

BGP MD5 Authentication on Fabric Links

XCO provides secure TCP using MD5 for BGP connections across all fabric links. You are able to configure or modify the MD5 password at any time.

This feature enables you to provide an `md5-password` as a fabric setting that further becomes configured on all the fabric links, that is, BGP peer-groups and individual neighbors without peer-groups on the SLX-OS devices, so that the peer sessions are established using MD5 authentication. This will be applied to both Clos and non-Clos fabrics. By default, MD5 authentication on the fabric is disabled. Any new fabric or the fabrics upgraded from previous releases will have the MD5 authentication disabled.

Configure BGP MD5 Password Create, Update, and Clear

Use the `efa fabric setting update` command to set or clear the MD5 password on a new fabric.

Here is the `efa fabric setting update` command:

```
efa fabric setting update --name <fabric-name> --md5-password-enable <yes/no> --md5-  
password <password>
```

If the command is entered with `md5-password-enable` as “yes” but without the `md5-password` option, then a prompt is displayed to input string and the password string entered using the prompt is not displayed on the screen.



Note

When providing a password string in the command line, that is using `efa fabric setting update --name <fabric-name> --md5-password <password>`, if the string contains special characters, then you must enclose the string in single quotes. For example, `efa fabric setting update --name fabric1 --md5-password 'pass%!'`. Enclosing the password string in single quotes is not required when the string is entered using the prompt.

After setting the `md5-password`, you must configure the fabric, using the command `efa fabric configure --name <fabric-name>`, to apply this MD5 password on fabric devices so that the BGP neighbor sessions are authenticated.

When you configure the `md5-password` on a fabric that has just been created, or a fabric that has not yet been configured, there is no change in the device `app-state`. However, if the `md5-password` is set after the fabric is configured, there is a new `app-state`, the fabric setting is refreshed (and devices will be set to), indicating the fabric properties have been modified and the fabric has to be reconfigured to apply the new settings. As part of fabric configuration, when the MD5 password was successfully configured on all the fabric links on a device, the app state on that device will go back to `cfg-in-sync` state.

1. Run the **`efa fabric setting update --name <fabric-name> --md5-password-enable <yes/no> --md5-password <password>`** command to set the MD5 password.

```
efa fabric setting update --name fabric1 --md5-password-enable yes  
Please supply a password for BGP MD5 authentication on fabric links:
```

2. Run the **`efa fabric configure --name <fabric-name>`** command to apply this MD5 password on fabric devices so that the BGP neighbor sessions are authenticated.

To create or update MD5 authentication:

```
efa fabric configure --name fabric1
```

To clear MD5 authentication:

```
efa fabric setting update --name fabric1 --md5-password-enable no  
efa fabric configure --name fabric1
```

When you configure the `md5-password` on a fabric that has just been created, or a fabric that has not yet been configured, there is no change in the device `app-state`. However, if the `md5-password` is set after the fabric is configured, the fabric status is set

to settings-updated along with the field BGP-MD5, indicating that settings have been updated. This indicates that the fabric properties have been modified and the fabric has to be reconfigured to apply the new settings. As part of fabric configure, when the devices are successfully configured, the fabric status will go back to configure-success.

Example

```
Fabric Name: fabric1, Fabric Description: , Fabric Stage: 3, Fabric Type: clos, Fabric
Status: settings-updated
Updated Fabric Settings: BGP-MD5
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
| IP ADDRESS | POD | HOST NAME | ASN | ROLE |
| DEVICE STATE | APP STATE | CONFIG GEN REASON | PENDING CONFIGS | VTLB ID | LB ID |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
| 10.17.112.223 | | spine1 | 64512 | spine | |
| provisioned | cfg in-sync | MD5 | MD5-U | NA | 1 |
| 10.17.112.224 | | spine2 | 64512 | spine |
| provisioned | cfg in-sync | MD5 | MD5-U | NA | 1 |
| 10.17.112.221 | | leaf1 | 65002 | leaf |
| provisioned | cfg in-sync | MD5 | MD5-U | 2 | 1 |
| 10.17.112.222 | | leaf2 | 65002 | leaf |
| provisioned | cfg in-sync | MD5 | MD5-U | 2 | 1 |
| 10.17.112.225 | | leaf3 | 65000 | leaf |
| provisioned | cfg in-sync | MD5 | MD5-U | 2 | 1 |
| 10.17.112.226 | | leaf4 | 65000 | leaf |
| provisioned | cfg in-sync | MD5 | MD5-U | 2 | 1 |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
FABRIC SETTING:
BGPLL - BGP Dynamic Peer Listen Limit, BGP-MD5 - BGP MD5 Password
CONFIG GEN REASON:
LD - Link Delete, LA - Link Add, IU - Interface
Update, PLC - IPPrefixList Create, PLD - IPPrefixList Delete, PLU - IPPrefixList Update
MD/MU - MCT Delete/Update, OD - Overlay Gateway Delete, OU - Overlay Gateway Update,
ED - Evpn Delete, PC - RouterPim Create, PD - RouterPim Delete, BGP - BGP Config
DD - Dependent Device Update, DA - Device Add, DR - Device ReAdd, ASN -
Asn Update, PU - RouterPim Update, SYS - System Properties Update, NA - Not Applicable
PENDING CONFIGS:
MCT - MCT Cluster, O - Overlay
Gateway, SYSP - System Properties, INTIP - Interface IP, BGP - BGP Config
C/D/U - Create/Delete/Update, PA/PD - Port Add/Port Delete
```

For App or Device Error/Failure reason, run "efa fabric error show" for details
For config refresh reason, run "efa fabric debug config-gen-reason" for details

**Note**

When the MD5 password is updated, for the new configuration to take effect, the neighbor sessions have to be cleared, resulting in a network outage until the new sessions are established. Because the configuration of the MD5 password toggles the network, a new warning message with a confirmation is provided indicating the impact of the `md5-password` setting on an active fabric, before it is applied. This warning message is displayed only when there is a need to reconfigure the fabric, that is, the password is set after the fabric is configured.

```
efa fabric setting modify --name fabric1 --md5-password-enable yes
```

```
Please supply a password for BGP MD5 authentication on fabric links:
```

```
WARNING: configuring/clearing md5-password on an active fabric will result in  
BGP neighborsessions  
going down for a brief period when the fabric is reconfigured.
```

```
Please confirm if you want to continue with the fabric setting update [y/n]?
```

Employ a Phased Approach

Reduce network impact by applying the password and clearing sessions on fabric links in a phased manner.

When an updated MD5 password is being applied on fabric links during fabric configuration, you can reduce network impact by applying the password and clearing sessions on fabric links in a phased manner. First gather a list of neighbor sessions on each device. Then, one device at a time, apply `md5-password` and clear a single peering. Application of password and clearing of the neighbor session is done on both ends of the peering session simultaneously. When the session is established, the MD5 password is applied on the next peering session. When all the neighbor sessions on the device are updated, it will move to the next device.

After you clear the session, it takes 10 seconds for the new session to be established. Before applying the new MD5 password, the session states are determined. After applying the password and clearing the session on both ends of the peering session, the session state is checked again. Only when the state matches with the previous (pre-password update) state on the session, or is better than the previous state (for example, previously the session was not established and the current state is established), it will move to updating the next neighbor session.

**Note**

The phased application of clearing the session and checking the state is performed **only** when the password is updated on a configured fabric and is not applicable during the configuration of a new fabric. Because the neighbor sessions are created for the first time during configuration of a new fabric with the MD5 password, there is no need to clear sessions.

Configure BGP MD5 Password: Failures When Clearing the Neighbor Session

When the MD5 password is updated, failures can occur during the process of clearing the neighbor sessions.

After clearing the session, if the session state is not established, or is not in the same state as it was prior to clearing the session within the wait time of 10 seconds, then the wait time is extended for an additional 10 seconds. If the session state is not established after the expiration of the second wait time, it is marked as a failure. The execution continues with the clearing of the remaining neighbor sessions.

When all the neighbor sessions are cleared, any sessions that have been marked as failure are presented under fabric errors at the end of the fabric configure operation, as part of the existing fabric error command `efa fabric error show`. The failure information will include the details of the neighbor session that could not be established.



Note

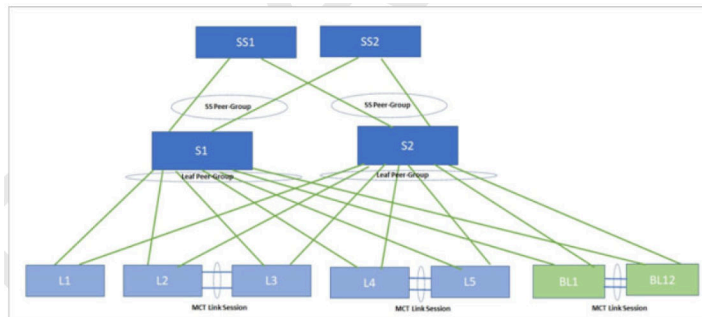
If there are any sessions that could not be established during the clear operation, the fabric configuration operation displays an error indicating a failure. However, if the fabric configuration has been successfully pushed to the devices, the devices are set to `cfg-in-sync`, even though the clear operation failed.

Configure BGP MD5 Password: Clos Topology (3-Stage and 5-Stage)

In a Clos topology, session clearing is done on each device on the fabric.

In a Clos topology, the clearing of the sessions is done by walking through each device on the fabric and the sessions cleared, depending on the role of the device. The clearing is done in the following manner:

- **Spine:** Neighbor sessions at the peer-group level are cleared. On the spine, there are two peer-groups, one for the links to the leaves (Leaf Peer-Group) and the other for the links to the super-spines (SS Peer-Group). Clearing of the sessions is done on both peer-groups simultaneously.
- **Leaf:** The neighbor session on the MCT link is cleared.
- **Super-spine:** There is no need to clear any session on the super-spine because all the sessions on the super-spines are covered at the spine device.



Configure BGP MD5 Password: Non-Clos Topology and Fabric Events

In a non-Clos topology, the process of clearing the session is performed on one leaf device at a time, until all sessions are cleared.

Because the MD5 password on fabric links is applied as part of fabric configure, no additional events are generated as a result of configuring MD5 password. The Fabric Deployed event that is currently generated by the Fabric service when a fabric is configured should be used as a trigger by other services that are dependent on the Fabric MD5 password for any of their operations.

Fabric Events and the MD5 Password

Because the MD5 password on each fabric link is applied as part of fabric configuration, no additional events are generated as a result of configuring the MD5 password.

The Fabric Deployed event that is generated by the fabric service when a fabric is configured should be used as a trigger by other services that are dependent on the fabric MD5 password for any of their operations.

Verify the BGP MD5 Password

Use the **efa fabric setting show** command to verify that the MD5 password is configured on the fabric.

The password is not displayed in clear text because of security concerns. The password is displayed as a hidden string (*****) or as an encrypted string. The password is displayed as a hidden string if the fabric is not configured after the MD5 password is set. And if the fabric is configured after the password is set, then the show command displays the password as an encrypted string. This encrypted string matches the password string displayed on the SLX-OS devices.

The same applies to the **efa show-running-config** command. Until the fabric is configured, the MD5 password is displayed as *****. After the fabric is configured, the encrypted string is displayed.

Run the **efa fabric setting show --name <fabric-name> --advanced** command.

```
efa fabric setting show --name fabric1 --advanced
```

NAME	VALUE
Fabric Name	fabric1
Link IP Range	10.10.10.0/23
Loopback IP Range	172.31.254.0/24
Loopback Port Number	1
VTEP Loopback Port Number	2
Spine ASN Block	64512-64768
SuperSpine ASN Block	64769
Leaf ASN Block	65000-65534

	Border Leaf ASN Block	66000-66100	
	P2P IP Type	numbered	
	Any cast MAC	0201.0101.0101	
	IPv6 Any cast MAC	0201.0101.0102	
	MAC Aging Timeout	1800	
	MAC Aging Conversational Timeout	300	
	MAC Move Limit	20	
	Duplicate MAC Timer	5	
	Duplicate MAC Timer MAX Count	3	
	BFD Enable	Yes	
	BFD Tx	300	
	BFD Rx	300	
	BFD Multiplier	3	
	BGP MultiHop	2	
	MaxPaths	8	
	AllowAsIn	0	
	MTU	9216	
	IPMTU	9100	
	MCT Link IP Range	10.20.20.0/24	
	MCT PortChannel	64	
	LACP Timeout	long	
	Control Vlan	4090	
	Control VE	4090	
	Leaf PeerGroup spine-group		
	Spine PeerGroup leaf-group		
	SuperSpine PeerGroup	spine-group	
	Configure Overlay Gateway	Yes	
	VNI Auto Map	Yes	
	Backup Routing Enable	No	
	Backup Routing IPv4 Range	10.40.40.0/24	
	Backup Routing IPv6 Range	fd40:4040:4040:1::/120	

Optimized Replication Enable	No	
+-----+-----+-----+		
MDT Group IPv4 Range	239.0.0.0/8	
+-----+-----+-----+		
Default MDT Group IPv4 address	239.1.1.1	
+-----+-----+-----+		
MD5 Password Enable	Yes	
+-----+-----+-----+		
MD5 Password	\$9\$GiXG/W7938rCj4lzgf14NQ==	
+-----+-----+-----+		

Configure BGP MD5 Password: Switch Configuration

The following provides a sample XCO MD5 password configuration and the corresponding switch configuration on one of the fabric devices after the fabric is configured.

1. Run the **efa fabric setting update** command.

```
efa fabric setting update --name fabric1 --md5-password-enable yes
Please supply a password for BGP MD5 authentication on fabric links:
efa fabric configure --name fabric1
```

2. Complete the following configuration on SLX-OS device:

```
router bgp
  local-as 65000
  capability as4-enable
  fast-external-fallover
  neighbor spine-group peer-group
  neighbor spine-group remote-as 64512
  neighbor spine-group description To Spine
  neighbor spine-group password $9$GiXG/W7938rCj4lzgf14NQ==
  neighbor 10.10.10.2 peer-group spine-group
  neighbor 10.20.20.2 remote-as 65000
  neighbor 10.20.20.2 next-hop-self
  neighbor 10.20.20.2 password $9$GiXG/W7938rCj4lzgf14NQ==
  address-family ipv4 unicast
  network 172.31.254.2/32
  maximum-paths 8
  graceful-restart
  !
  address-family ipv6 unicast
  !
  address-family l2vpn evpn
  graceful-restart
  neighbor spine-group encapsulation vxlan
  neighbor spine-group next-hop-unchanged
  neighbor spine-group enable-peer-as-check
  neighbor spine-group activate
  !
  !
```

The BGP MD5 Password, Drift and Reconcile, and Idempotency

Drift is identified if you modify the MD5 password through SLX-OS , the CLI, or other management tool.

A reconcile operation pushes the intended configuration to SLX-OS , thereby synchronizing the SLX-OS configuration with XCO.



Note

A reconcile operation configures the MD5 password on the device back to its original value (pre-drift) but does not clear the session. Also, the state is not verified after the password is configured.

Field	Identity Drift	Reconcile Configuration	Idempotency
md5-password	Yes	Yes	Yes*

* There are some caveats to idempotency. The fabric service does not store the plain text password you provide after the fabric has been configured. It stores the encrypted string of the user-provided password, matching with the encrypted string available on the SLX-OS device. So, setting the same original MD5 password after the fabric is configured results in devices going into cfg-refreshed state. For the operation to be idempotent, after the fabric is configured, the encrypted string should be provided as the `md5-password` and not the original plain text password.

Devices in the fabric are in cfg-refreshed state when the MD5 password has been updated but the fabric is not yet reconfigured. In such a scenario, the previous MD5 password is used for drift detection until the fabric is configured with the new password.

Configure BGP MD5 Password: Fabric Clone

When a fabric is cloned using the command `efa fabric clone --source <old-fabric-name> -- destination <new-fabric-name>`, the MD5 password configuration from the source fabric is used in the new cloned fabric.

Configure BGP MD5 Password: Rules for Clearing BGP Sessions

MD5 password configuration and clearing BGP neighbor sessions.

The following table lists different scenarios of MD5 password configuration, under which a BGP neighbor session is required to be cleared.

Scenarios	Clearing BGP neighbor session required?
Creating new BGP peer-group with MD5 password	Not required
Creating new BGP neighbor with MD5 password	Not required
Updating existing BGP peer-group with MD5 password	Yes

Scenarios	Clearing BGP neighbor session required?
Updating existing BGP neighbor with MD5 password	Yes
Removing MD5 password from a BGP peer-group	Yes
Removing MD5 password from a BGP neighbor	Yes
Modifying MD5 password for a BGP peer-group	Yes
Modifying MD5 password for a BGP neighbor	Yes
Reload	Not required
Copy <ftp://backup-config> startup-config and reload	Not required
Copy <ftp://backup-config> running-config	Not required

SLX-OS Commands to clear the BGP neighbor sessions

```
clear ip bgp neighbor <neighbor ip> vrf <vrf name>
```

```
clear ip bgp neighbor <peer-group> vrf <vrf name>
```

The `vrf` used on the XCO fabric links is `default-vrf`.

BGP MD5 Authentication on Edge Links

This feature authenticates all the BGP peer and peer-group used for edge connectivity. You can provide an MD5 password per BGP peer and peer-group created for external connectivity.



Important

BGP MD5 authentication for tenant dynamic peers is not yet supported.

Configure BGP MD5 Authentication for Tenant BGP Peer

Provide `md5-password` during BGP peer create or update operations.

1. Run the **efa tenant service bgp peer create** command to create the peer.

```
efa tenant service bgp peer create
--name <bgp-peer-name> --tenant <tenant-name>
--ipv4-uc-nbr <device-ip,vrf-name:neighbor-ip,remote-asn>
--ipv4-uc-nbr-bfd <device-ip,vrf-name:neighbor-ip,true|false>
--ipv4-uc-nbr-md5-password <device-ip,vrf-name:neighborip, ipv4-md5-
password>
```

2. Run the **efa tenant service bgp peer update** command to update the peer.

```
efa tenant service bgp peer update
--name <bgp-peer-name> --tenant <tenant-name>
--operation peer-add
```

```

--ipv4-uc-nbr <device-ip,vrf-name:neighbor-ip,remote-asn>
--ipv4-uc-nbr-bfd <device-ip,vrf-name:neighbor-ip,true|false>
--ipv4-uc-nbr-md5-password <device-ip,vrf-name:neighborip, ipv4-md5-
password>

```

Example

```

efa tenant service bgp peer create
--name tenlbgppeer1 --tenant ten1
--ipv4-uc-nbr 10.20.246.15,ten1vrf1:10.20.30.40,50000
--ipv4-uc-nbr-bfd 10.20.246.15,ten1vrf1:10.20.30.40,true
--ipv4-uc-nbr-md5-password 10.20.246.15,ten1vrf1:10.20.30.40,password
--ipv4-uc-nbr 10.20.246.16,ten1vrf1:10.20.30.40,50000
--ipv4-uc-nbr-bfd 10.20.246.16,ten1vrf1:10.20.30.40,true
--ipv4-uc-nbr-md5-password 10.20.246.16,ten1vrf1:10.20.30.40,password
efa tenant service bgp peer update
--name tenlbgppeer1 --tenant ten1
--operation peer-add
--ipv4-uc-nbr 10.20.246.15,ten1vrf1:10.20.30.50,50000
--ipv4-uc-nbr-bfd 10.20.246.15,ten1vrf1:10.20.30.50,true
--ipv4-uc-nbr-md5-password 10.20.246.15,ten1vrf1:10.20.30.50,password1
--ipv4-uc-nbr 10.20.246.16,ten1vrf1:10.20.30.50,50000

```

```
--ipv4-uc-nbr-bfd 10.20.246.16,ten1vrfl:10.20.30.50,true
--ipv4-uc-nbr-md5-password 10.20.246.16,ten1vrfl:10.20.30.50,password1
```

efa tenant service bgp peer show --detail		Device IP :	
=====		10.20.246.16	
Name : ten1bgppeer1		VRF : ten1vrfl	
Tenant : ten1		AFI : ipv4	
State : bs-state-created		SAFI : unicast	
Description :		Remote IP :	
Static Peer		10.20.30.40	
-----		Remote ASN : 50000	
Device IP :		Next Hop Self : false	
10.20.246.15		Update Source IP :	
VRF : ten1vrfl		BFD Enabled : true	
AFI : ipv4		BFD Interval : 0	
SAFI : unicast		BFD Rx : 0	
Remote IP :		BFD Multiplier : 0	
10.20.30.40		MD5 Password :	
Remote ASN : 50000		\$9\$MCgKGaNT6OASX68/7TC6Lw==	
Next Hop Self : false		Dev State :	
Update Source IP :		provisioned	
BFD Enabled : true		App State : cfg-in-	
BFD Interval : 0		sync	
BFD Rx : 0		Device IP :	
BFD Multiplier : 0		10.20.246.16	
MD5 Password :		VRF : ten1vrfl	
\$9\$MCgKGaNT6OASX68/7TC6Lw==		AFI : ipv4	
Dev State :		SAFI : unicast	
provisioned		Remote IP :	
App State : cfg-in-		10.20.30.50	
sync		Remote ASN : 50000	
Device IP :		Next Hop Self : false	
10.20.246.15		Update Source IP :	
VRF : ten1vrfl		BFD Enabled : true	
AFI : ipv4		BFD Interval : 0	
SAFI : unicast		BFD Rx : 0	
Remote IP :		BFD Multiplier : 0	
10.20.30.50		MD5 Password :	
Remote ASN : 50000		\$9\$ufD04Gw+49ex4H8UtvifqA==	
Next Hop Self : false		Dev State :	
Update Source IP :		provisioned	
BFD Enabled : true		App State : cfg-in-	
BFD Interval : 0		sync	
BFD Rx : 0		Dynamic Peer	
BFD Multiplier : 0		-----	
MD5 Password :		0 Records	
\$9\$ufD04Gw+49ex4H8UtvifqA==		0 Records	
Dev State :		=====	
provisioned		=====	
App State : cfg-in-			
sync			

3. Complete the configuration on SLX-OS as provided in the following example.

<pre> L1# show running-config router bgp router bgp local-as 42000000000 capability as4-enable fast-external-fallover neighbor 10.20.20.4 remote-as 4200000000 neighbor 10.20.20.4 next-hop-self address-family ipv4 unicast network 172.31.254.46/32 network 172.31.254.123/32 maximum-paths 8 graceful-restart ! address-family ipv4 unicast vrf tenlvrf1 redistribute connected neighbor 10.20.30.40 remote-as 50000 neighbor 10.20.30.40 password \$9\$MCgKGaT6OASX68/7TC6Lw== neighbor 10.20.30.40 bfd neighbor 10.20.30.50 remote-as 50000 neighbor 10.20.30.50 password \$9\$ufD04Gw+49ex4H8UtvifqA== neighbor 10.20.30.50 bfd maximum-paths 8 ! address-family ipv6 unicast ! address-family ipv6 unicast vrf tenlvrf1 redistribute connected maximum-paths 8 ! address-family l2vpn evpn graceful-restart ! ! </pre>	<pre> L2# show running-config router bgp router bgp local-as 42000000000 capability as4-enable fast-external-fallover neighbor 10.20.20.5 remote-as 4200000000 neighbor 10.20.20.5 next-hop-self address-family ipv4 unicast network 172.31.254.46/32 network 172.31.254.176/32 maximum-paths 8 graceful-restart ! address-family ipv4 unicast vrf tenlvrf1 redistribute connected neighbor 10.20.30.40 remote-as 50000 neighbor 10.20.30.40 password \$9\$MCgKGaT6OASX68/7TC6Lw== neighbor 10.20.30.40 bfd neighbor 10.20.30.50 remote-as 50000 neighbor 10.20.30.50 password \$9\$ufD04Gw+49ex4H8UtvifqA== neighbor 10.20.30.50 bfd maximum-paths 8 ! address-family ipv6 unicast ! address-family ipv6 unicast vrf tenlvrf1 </pre>
---	---



Note

The MD5 password cannot be set or unset on an existing BGP peer present within a peer instance. You need to remove the BGP peer from the BGP peer instance and then add back the BGP peer to the peer instance with the desired MD5 password configuration.

Configure BGP MD5 Authentication for Tenant BGP Peer-group

You can provide an MD5 password during BGP peer-group create or update operations.

1. Use the **efa tenant service bgp peer-group create** command to create the peer group.

```

efa tenant service bgp peer-group create
  --name <bgp-pg-name> --tenant <tenant-name>
  --pg-name <device-ip:pg-name> --pg-asn <device-ip,pg-name:remote-asn>
  --pg-bfd-enable <device-ip,pg-name:true|false>
  --pg-md5-password <device-ip,pg-name:md5-password>

```

2. Use the **efa tenant service bgp peer-group update** command to update the peer group.

```
efa tenant service bgp peer-group update
--name <bgp-pg-name> --tenant <tenant-name>
--operation peer-group-add
--pg-name <device-ip:pg-name> --pg-asn <device-ip,pg-name:remote-asn>
--pg-bfd-enable <device-ip,pg-name:true|false>
--pg-md5-password <device-ip,pg-name:md5-password>
```

Example

```
efa tenant service bgp peer-group create
--name ten1bgppg1 --tenant ten1
--pg-name 10.20.246.15:pg1 --pg-asn 10.20.246.15,pg1:55001
--pg-bfd-enable 10.20.246.15,pg1:true
--pg-md5-password 10.20.246.15,pg1:password
--pg-name 10.20.246.16:pg1 --pg-asn 10.20.246.16,pg1:55001
--pg-bfd-enable 10.20.246.16,pg1:true
--pg-md5-password 10.20.246.16,pg1:password
```

```
efa tenant service bgp peer-group update
--name ten1bgppg1 --tenant ten1
--operation peer-group-add
--pg-name 10.20.246.15:pg2 --pg-asn 10.20.246.15,pg2:55002
--pg-bfd-enable 10.20.246.15,pg2:true
--pg-md5-password 10.20.246.15,pg2:password1
--pg-name 10.20.246.16:pg2 --pg-asn 10.20.246.16,pg2:55002
--pg-bfd-enable 10.20.246.16,pg2:true
--pg-md5-password 10.20.246.16,pg2:password1
```

```
efa tenant service bgp peer-group show --detail
```

```
=====
=====
Name                : ten1bgppg1
Tenant              : ten1
State               : bgp-pg-state-created

Peer Group
-----
Device IP           : 10.20.246.15
Peer Group          : pg1
Remote ASN          : 55001
Next Hop Self       : false
BFD Enabled         : true
BFD Interval        :
BFD Rx              :
BFD Multiplier      :
MD5 Password        : $9$MCgKGaT6OASX68/7TC6Lw==
Dev State           : provisioned
App State            : cfg-in-sync

Device IP           : 10.20.246.15
Peer Group          : pg2
Remote ASN          : 55002
Next Hop Self       : false
BFD Enabled         : true
BFD Interval        :
BFD Rx              :
BFD Multiplier      :
MD5 Password        : $9$ufD04Gw+49ex4H8UtvifqA==
```

```

Dev State      : provisioned
App State      : cfg-in-sync

Device IP      : 10.20.246.16
Peer Group     : pg1
Remote ASN     : 55001
Next Hop Self  : false
BFD Enabled    : true
BFD Interval   :
BFD Rx         :
BFD Multiplier :
MD5 Password : $9$MCgKGaT6OASX68/7TC6Lw==
Dev State      : provisioned
App State      : cfg-in-sync

Device IP      : 10.20.246.16
Peer Group     : pg2
Remote ASN     : 55002
Next Hop Self  : false
BFD Enabled    : true
BFD Interval   :
BFD Rx         :
BFD Multiplier :
MD5 Password : $9$ufD04Gw+49ex4H8UtvifqA==
Dev State      : provisioned
App State      : cfg-in-sync

```

```

=====
=====

```

3. Complete the following configuration on SLX-OS .

<pre> L1# show running-config router bgp router bgp local-as 4200000000 capability as4-enable fast-external-fallover neighbor pg1 peer-group neighbor pg1 remote-as 55001 neighbor pg1 password \$9\$MCgKGaT6OASX68/7TC6Lw== neighbor pg1 bfd neighbor pg2 peer-group neighbor pg2 remote-as 55002 neighbor pg2 password \$9\$ufD04Gw+49ex4H8UtvifqA== neighbor pg2 bfd neighbor 10.20.20.4 remote-as 4200000000 neighbor 10.20.20.4 next-hop-self address-family ipv4 unicast network 172.31.254.46/32 network 172.31.254.123/32 maximum-paths 8 graceful-restart ! address-family ipv4 unicast vrf tenlvrf1 redistribute connected maximum-paths 8 ! address-family ipv6 unicast ! address-family ipv6 unicast vrf tenlvrf1 redistribute connected maximum-paths 8 ! address-family l2vpn evpn graceful-restart ! ! </pre>	<pre> L2# show running-config router bgp router bgp local-as 4200000000 capability as4-enable fast-external-fallover neighbor pg1 peer-group neighbor pg1 remote-as 55001 neighbor pg1 password \$9\$MCgKGaT6OASX68/7TC6Lw== neighbor pg1 bfd neighbor pg2 peer-group neighbor pg2 remote-as 55002 neighbor pg2 password \$9\$ufD04Gw+49ex4H8UtvifqA== neighbor pg2 bfd neighbor 10.20.20.5 remote-as 4200000000 neighbor 10.20.20.5 next-hop-self address-family ipv4 unicast network 172.31.254.46/32 network 172.31.254.176/32 maximum-paths 8 graceful-restart ! address-family ipv4 unicast vrf tenlvrf1 redistribute connected maximum-paths 8 ! address-family ipv6 unicast ! address-family ipv6 unicast vrf tenlvrf1 redistribute connected maximum-paths 8 ! address-family l2vpn evpn graceful-restart ! ! </pre>
--	--



Note

The MD5 password cannot be set or unset on an existing BGP peer-group present within a peer-group instance. You need to remove the BGP peer-group from the BGP peer-group instance and then add back the BGP peer-group to the peer-group instance with the desired MD5 password configuration.

Configure BGP MD5 Authentication for Tenant BGP Peer and Peer-group Securely

You can securely provide MD5 passwords during BGP peer-group create or update operations.

[Configure BGP MD5 Authentication for Tenant BGP Peer](#) on page 38 and [Configure BGP MD5 Authentication for Tenant BGP Peer-group](#) on page 41 present instructions for providing an md5-password per BGP peer or peer-group during the BGP peer or peer-group create and update operations.

This topic provides an additional method for doing so – in a secure manner – using the `--md5-password-prompt-enable=true` option in the **efa tenant service bgp peer create** and **efa tenant service bgp peer-group create** commands.

You are prompted to supply a password the same number of times as the number of BGP peer or peer-group inputs you specify in the command. Then, you can choose to type in a password, in which case, it is not shown. Alternatively, you can skip the input of the password altogether by pressing Enter.



Note

You can choose to provide the BGP peer or peer-group md5-password either in a secure manner (using the prompt) or in an unsecure manner, as described in the topics referenced above, but not both.

1. Run the **efa tenant service bgp peer create** command to create or update a BGP peer.

For example:

```
efa tenant service bgp peer create --name bgp173-2501 --tenant tenant11
--ipv4-uc-nbr 10.20.246.6,v1:25.1.1.3,5901
--ipv4-uc-nbr-bfd 10.20.246.6,v1:25.1.1.3,true
--ipv6-uc-nbr 10.20.246.5,v1:25:1::3,5901
--ipv6-uc-nbr-bfd 10.20.246.5,v1:25:1::3,true
--md5-password-prompt-enable=true
```

The following output is displayed:

```
Enter Md5 Password for 10.20.246.6::v1::25.1.1.3:
```

For the first prompt, suppose you enter a password. It is not displayed on the screen. You are prompted for the second password:

```
Enter Md5 Password for 10.20.246.5::v1::25:1::3:
```

Suppose that this time, you do not enter a password at all but press Enter.

The following output is displayed:

BgpService created successfully.

2. Run the **efa tenant service bgp peer-group create** command to create or update a BGP peer-group.

For example:

```
efa tenant service bgp peer-group create --tenant "tenant11" --name "v1-PeerGrp"
--pg-name 10.20.246.5:v1-PeerGrp --pg-asn 10.20.246.5,v1-
PeerGrp:5200
--pg-bfd-enable 10.20.246.5,v1-PeerGrp:true
--pg-name 10.20.246.5:v3-PeerGrp --pg-asn 10.20.246.5,v3-
PeerGrp:5201
--pg-bfd-enable 10.20.246.5,v3-PeerGrp:true
--pg-name 10.20.246.6:v1-PeerGrp --pg-asn 10.20.246.6,v1-
PeerGrp:5200
--pg-bfd-enable 10.20.246.6,v1-PeerGrp:true
--pg-md5-password-prompt-enable=true
```

The following output is displayed:

```
Enter Md5 Password for 10.20.246.5::v1-PeerGrp:
```

For the first prompt, suppose you enter a password. It is not displayed on the screen.

You are prompted for the second password:

```
Enter Md5 Password for 10.20.246.5::v3-PeerGrp:
```

Suppose you enter a password this time, too. It is not displayed on the screen.

Now you are prompted a third time:

```
Enter Md5 Password for 10.20.246.6::v1-PeerGrp:
```

Suppose that this time, you do not type in a password at all but press Enter.

The following output is displayed:

```
BgpService created successfully.
```

Configure BGP MD5 Authentication for Backup Routing Neighbors

The BGP MD5 password for the backup routing neighbors is the same as the one set at the fabric setting level. The BGP MD5 password for the backup routing neighbors is configured during the configuration of VRF on SLX-OS.

If the MD5 password setting is updated or set on a provisioned fabric followed by **efa fabric configure**, then the modified backup routing neighbor configuration is applied on all the tenant VRF backup routing BGP neighbors during **efa fabric configure**.

1. Run the series of commands as shown in the following example.

```
efa fabric show --name fabric1
Fabric Name: fabric1, Fabric Description: , Fabric Type: non-clos
+-----+-----+-----+-----+-----+-----+-----+
| IP ADDRESS | RACK | HOST NAME | ASN | ROLE | DEVICE STATE | APP STATE |
| CONFIG GEN REASON | PENDING CONFIGS | VTLB ID | LB ID |
+-----+-----+-----+-----+-----+-----+-----+
| 10.20.246.15 | rack1 | Avalanche-01 | 4200000000 | leaf | provisioned | cfg in-sync |
| NA | NA | | 2 | 1 |
| 10.20.246.16 | rack1 | Avalanche-02 | 4200000000 | leaf | provisioned | cfg in-sync |
| NA | NA | | 2 | 1 |
+-----+-----+-----+-----+-----+-----+-----+
efa fabric setting show --name fabric1 --advanced | grep -i "backup routing"
| Backup Routing Enable | Yes |
| Backup Routing IPv4 Range | 10.40.40.0/24 |
| Backup Routing IPv6 Range | fd40:4040:4040:1::/120 |

efa tenant show
+-----+-----+-----+-----+-----+-----+-----+
| Name | Type | VLAN Range | L2VNI Range | L3VNI Range | VRF Count | Enable BD |
| Ports |
+-----+-----+-----+-----+-----+-----+-----+
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| ten1 | private | 11-20 | 20001-20020 | 21001-210020 | 10 | false |
| 10.20.246.15[0/1-10] |
| | | | | | |
| 10.20.246.16[0/1-10] |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+

efa tenant vrf show
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| Name | Tenant | Routing Type | Centralized Routers | Redistribute | Max Path |
| Local Asn | Enable GR | State | Dev State | App State |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| ten1vrf1 | ten1 | distributed | | connected | 8
| | false | vrf-create | not-provisioned | cfg-ready |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+

efa fabric setting show --name fabric1 -advanced | grep -i MD5
| MD5 Password Enable | Yes
| MD5 Password | $9$jrujIQqNxWkAyUOoI4cMtzhc4oP2VGREKwLOsSKH8bw= |

efa tenant epg show --name tenlepg1 --tenant ten1 -detail
=====
Name : tenlepg1
Tenant : ten1
Type : extension
State :
Description :
Ports : 10.20.246.15[0/1]
POs :
Port Property : SwitchPort Mode : trunk
: Native Vlan Tagging : false
: Single-Homed BFD Session Type : auto
NW Policy : Ctag Range : 11
: VRF : ten1vrf1
: L3Vni : 21001
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
+-----+
+-----+
| Ctag | Ctag | L2Vni | BD Name | Anycast IPv4 | Anycast IPv6
| Local IP | IP MTU | IPv6 ND | IPv6 ND | IPv6 ND | Dev
State | App State |
| Description | | | | |
[Device-IP->Local-IP] | Mtu | Managed Config | Other Config |
| |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
+-----+
| 11 | Tenant L3 Extended VLAN | 20001 | | 10.0.11.1/24 |
| | | | false | false |
provisioned | cfg-in-sync |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
+-----+

```

2. Complete the configuration on SLX-OS as provided in the following example.

<pre> L1# show running-config router bgp router bgp local-as 4200000000 capability as4-enable fast-external-fallover neighbor 10.20.20.3 remote-as 4200000000 neighbor 10.20.20.3 next-hop-self address-family ipv4 unicast network 172.31.254.71/32 network 172.31.254.151/32 maximum-paths 8 graceful-restart ! address-family ipv4 unicast vrf tenlvrf1 redistribute connected neighbor 10.40.40.252 remote-as 4200000000 neighbor 10.40.40.252 next-hop- self neighbor 10.40.40.252 password \$9\$jrujIQqNxWkAyUOoI4cMtzhc4oP2VGRE KwLOsSKH8bw= maximum-paths 8 ! address-family ipv6 unicast ! address-family ipv6 unicast vrf tenlvrf1 redistribute connected neighbor fd40:4040:4040:1::fe remote-as 4200000000 neighbor fd40:4040:4040:1::fe next-hop-self neighbor fd40:4040:4040:1::fe password \$9\$jrujIQqNxWkAyUOoI4cMtzhc4oP2VGRE KwLOsSKH8bw= neighbor fd40:4040:4040:1::fe activate maximum-paths 8 ! address-family l2vpn evpn graceful-restart ! ! </pre>	<pre> L2# show running-config router bgp router bgp local-as 4200000000 capability as4-enable fast-external-fallover neighbor 10.20.20.2 remote-as 4200000000 neighbor 10.20.20.2 next-hop-self address-family ipv4 unicast network 172.31.254.71/32 network 172.31.254.195/32 maximum-paths 8 graceful-restart ! address-family ipv4 unicast vrf tenlvrf1 redistribute connected neighbor 10.40.40.253 remote-as 4200000000 neighbor 10.40.40.253 next-hop- self neighbor 10.40.40.253 password \$9\$jrujIQqNxWkAyUOoI4cMtzhc4oP2VGRE KwLOsSKH8bw= maximum-paths 8 ! address-family ipv6 unicast ! address-family ipv6 unicast vrf tenlvrf1 redistribute connected neighbor fd40:4040:4040:1::ff remote-as 4200000000 neighbor fd40:4040:4040:1::ff next-hop-self neighbor fd40:4040:4040:1::ff password \$9\$jrujIQqNxWkAyUOoI4cMtzhc4oP2VGRE KwLOsSKH8bw= neighbor fd40:4040:4040:1::ff activate maximum-paths 8 ! address-family l2vpn evpn graceful-restart ! ! </pre>
--	--

XCO Certificate Management

The following certificates in XCO are automatically generated during installation and registration of devices:

- Device Certificates
- XCO Certificates

Device Certificates

Device certificates are installed and configured during the SLX and Extreme OS ONE device registration in XCO.

During the registration of an SLX device in XCO, the following certificates are installed on the device:

1. **OAuth Certificate:** The public certificate for verifying an XCO token is copied to the device. This is the JWT Certificate described in XCO Certificates.
2. **Syslog Certificate:** To push messages to XCO over port 6514.
3. **HTTPS Certificate:** To enable secure communication with the clients.

When registering a Extreme OS ONE device in XCO, the following certificates are installed on the device:

1. **GRPC Certificate:** To enable secure communication with the clients.
2. **Syslog Certificate:** To push messages to XCO over port 6514.
3. **JWT Certificate:** The public certificate for verifying an XCO token is copied to the device.

Along with the certificate installation, the following configuration changes are done on the registered SLX device:

1. HTTP mode is disabled on the device, and HTTPS is enabled.
2. OAuth2 is enabled as the primary mode of authentication. Fallback is set to `local login`.

Along with the certificate installation, the following configuration change is done on the registered OS ONE device:

1. Configure the `grpc-server`
2. Assign the certificate ID

Use the `efa inventory device list` command to verify the status of the certificates on the device. If the **Cert/Key Saved** column contains "N," then certificates are not installed.

Syslog CA

Use this topic to learn about the third-party certificates for RASlog service (syslog from SLX).

XCO is shipped with default certificates. These are self-signed and the same certificates are used for listening to the syslog messages received from SLX.

```
$ efa inventory device register --ip=10.x.x.x --username=<username> --password=<password>
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | IP Address | Host Name | Model | Chassis Name | Firmware | Status | Reason |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 10.x.x.x | SLX | 3012 | SLX9250-32C | 20.2.3d | Success | |
+-----+-----+-----+-----+-----+-----+-----+-----+
Device Details
--- Time Elapsed: 1m6.570042048s ---
```

The syslog certificate on the device is the default CA that XCO contains. XCO Intermediate CA is pushed to SLX for mutual TLS over 6514 port to receive messages from SLX.

```
SLX# show crypto ca certificates
syslog CA certificate(Server authentication):
```

```

SHA1 Fingerprint=A3:E8:F6:CB:46:F6:43:C5:D1:90:1F:A7:C6:58:93:29:77:6F:2F:8E
Subject: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation Intermediate,
CN=EFA Intermediate CA/emailAddress=support@extremenetworks.com
Issuer: C=US, ST=CA, L=SJ, O=Extreme Networks, OU=Extreme Fabric Automation,
CN=efa.extremenetworks.com/emailAddress=support@extremenetworks.com
Not Before: Feb 20 22:25:26 2020 GMT
Not After : Feb 17 22:25:26 2030 GMT

```

An enhancement updates RASlog service to use the custom certificates that XCO servers use. The certificate CLI on XCO contains a new parameter, which enables you to upload CA.

```

$ efa certificate server --certificate=my_server_162.pem --key=my_server_162.key --
cacert=ca-chain.pem
Please wait as the certificates are being installed...
Certificates were installed!
--- Time Elapsed: 30.946303683s ---

```

If a third-party certificate is installed on XCO along with CA, syslog CA will be pushed to the device instead of the default XCO Intermediate CA.

```

SLX# show crypto ca certificates
syslog CA certificate(Server authentication):
SHA1 Fingerprint=32:70:EB:91:F4:6D:9C:9F:6E:35:E0:00:20:B8:1A:FF:AF:BA:0D:8A
Subject: C=US, O=xyz, OU=abcd, CN=INTERIM-CN
Issuer: C=US, O=xzy, OU=abcd, CN=ROOT-CN
Not Before: Feb 15 14:56:08 2022 GMT
Not After : Nov 11 14:56:08 2024 GMT

```

If you do not provide any CA certificate, the default certificates of XCO are used. If there are already registered devices, then the syslog certificate is automatically updated on these devices.

Expiry and Alerts

Syslog CA has the same expiry as of XCO Intermediate CA or the third-party CA. Legacy notification is sent to the users in case the certificate is going to expire in 30 days. It supports the following alerts which effects the health of XCO security subsystem.

- DeviceCertificateExpiryNoticeAlert
- DeviceCertificateExpiredAlert
- DeviceCertificateUnreadableAlert

For more information, see *Fault Management – Alerts* topic in the *ExtremeCloud™ Orchestrator CLI Administration Guide*.

Renewal

Performing a manual device update or DRC will also update the SLX syslog certificate.

- Manual Device Update

```
efa inventory device update (--ip [ip_address] | --fabric [fabric_name])
```

- Manual Drift Reconcile

```
efa inventory drift-reconcile execute --ip [ip_address] --reconcile
```

OAuth Certificate

JWT Verifier from XCO is pushed to SLX during registration.

```
SLX# show crypto ca certificates
oauth2 certificate(OAuth2 token signature validation):
SHA1 Fingerprint=57:55:2F:7A:F0:DB:23:CF:37:67:8D:AE:82:35:D8:2D:18:00:17:9E
Subject: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation,
CN=extremenetworks.com
Issuer: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation,
CN=extremenetworks.com
Not Before: Sep 2 13:26:27 2022 GMT
Not After : Aug 30 13:26:27 2032 GMT
```

Expiry and Alerts

Legacy notification is sent to the user if the certificate is going to expire in 30 days. It supports the following alerts which effects the health of XCO security subsystem:

- DeviceCertificateExpiryNoticeAlert
- DeviceCertificateExpiredAlert
- DeviceCertificateUnreadableAlert

For more information, see [Fault Management – Alerts](#).

Upload or Renewal

To upload the token signing certificate to the device, run the following command:

```
(efa:extreme)extreme@tpvm:~$ efa certificate device install --ip=10.x.x.x --certtype=
token
+-----+-----+
| IP Address | Status |
+-----+-----+
| 10.x.x.x   | Success |
+-----+-----+
---Time Elapsed: 27.233017418s ---
```

For more information about updating the certificates, see [Manual Installation of Certificates on Devices](#) on page 53.

On renewal of certificate, CertificateRenewalAlert is raised which changes the health of the system to green.

HTTPS Certificates

When you register a SLX device in XCO, a new certificate is generated for the HTTPS server of SLX device. The certificate is generated with the default CA that XCO contains.

The following is an example of a certificate on SLX after device registration:

```
slx-171# show crypto ca certificates

Certificate Type: https; Trustpoint: none
certificate:
SHA1 Fingerprint=C1:F1:2C:BF:1A:47:7B:46:5D:8F:18:99:0E:58:CF:31:8C:58:5F:CC
Subject: CN=slx-10.x.x.x.extremenetworks.com
Issuer: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation Intermediate,
CN=EFA Intermediate CA/emailAddress=support@extremenetworks.com
```

```
Not Before: Jan 10 11:12:18 2022 GMT
Not After : Jan 10 11:12:18 2024 GMT
```

You can use the CLI command only to install third-party certificates on a single device at once.

```
(efa:extreme)extreme@tpvm:/apps/test/certs$ efa certificate device install --ip=10.x.x.x
--cert-type https --https-certificate server.crt --https-key
my_server.key

WARNING: This will restart the HTTP service on the SLX devices and GRPC server on the OS
ONE devices. The services will not be able to connect till the operation is complete. Do
you want to proceed [y/n]?

y
+-----+-----+
| IP Address | Status |
| 10.20.61.171 | Success |
+-----+-----+
--- Time Elapsed: 38.516844258s ---
```

The device must have the new certificates uploaded:

```
slx-171# show crypto ca certificates

Certificate Type: https; Trustpoint: none
certificate:
SHA1 Fingerprint=D8:49:5F:12:AC:FE:BB:CB:95:C2:AC:6B:AF:B6:5B:9E:24:66:59:7D
Subject: CN=10.x.x.x/subjectAltName=IP=10.20.61.171
Issuer: C=US, O=xyz, OU=abcd, CN=INTERIM-CN
Not Before: Feb 10 11:23:36 2022 GMT
Not After : Jun 25 11:23:36 2023 GMT
```

Expiry and Alerts

The HTTPS certificate generated for SLX has an expiry of two years from the date of registration. The device shows the following error message when an HTTP certificate expires:

```
1022 AUDIT, 2025/06/24-17:20:52 (GMT), [SEC-3112], INFO, SECURITY, admin/admin/127.0.0.1/
http/REST Interface,, SLX, Event: X509v3, Certificate Validation failed, Info: Reason =
certificate has expired,
Certificate Details = [Subject CN efa.extremenetworks.com,
Serial 16193545342960822577 Issuer /C=US/ST=CA/O=Extreme Networks/OU=Extreme Fabric
Automation Intermediate/CN=EFA Intermediate CA/emailAddress=support@extremenetworks.com].
```

Legacy notification is sent to the users if the certificate is going to expire in 30 days. It supports the following alerts which effects the health of XCO security subsystem:

```
- DeviceCertificateExpiryNoticeAlert
- DeviceCertificateExpiredAlert
- DeviceCertificateUnreadableAlert
```

For more information, see **Fault Management - Alerts** topic in the ExtremeCloud™ Orchestrator CLI Administration Guide.

Upload or Renewal

To upload the HTTPS certificate to the device, use the following command:

```
(efa:extreme)extreme@tpvm:~$ efa certificate device install --ip=10.x.x.x --certtype=
https

WARNING: This will restart the HTTP service on the SLX devices and GRPC server on the OS
ONE devices. The services will not be able to connect till the operation is complete. Do
```

```

you want to proceed [y/n]?
y
+-----+-----+
| IP Address | Status |
+-----+-----+
| 10.x.x.x   | Success |
+-----+-----+
---Time Elapsed: 27.233017418s ---

```

For more information about updating the certificates, see [Manual Installation of Certificates on Devices](#) on page 53.

On renewal of certificate, `CertificateRenewalAlert` is raised which changes the health of the system to green.

GRPC Certificates

When you register an OS ONE device in XCO, a new certificate is generated for the GRPC server of the OS ONE device. The certificate is generated with the default CA that XCO contains. You can use the CLI command to install third-party certificates on a single OS ONE device at once.

```
$ efa certificate device install --ip 10.x.x.x --grpc-certificate device1.pem --grpc-key device1.key
```

```

WARNING: This will restart the HTTP service on the SLX devices and GRPC server on the OS
ONE devices. The services will not be able to connect till the operation is complete. Do
you want to proceed [y/n]?

```

```

Do you want to proceed [y/n]
+-----+-----+
| IP Address | Status |
+-----+-----+
| 10.x.x.x   | Success |
+-----+-----+

```

Manual Installation of Certificates on Devices

You can upload HTTPS and Token certificate on the devices using the following command:

```

efa certificate device install --help
Install certificates on devices

Usage:
  efa certificate device install [flags]

Flags:
  --ip string           Comma separated range of device IP addresses.
                        Example: 1.1.1.1-3,1.1.1.2,2.2.2.2
  --fabric string       Specify the name of the fabric
  --cert-type string    Certificate Type (https | token)
  --https-certificate string Local path to the certificate pem file
  --https-key string    Local path to the key pem file
  --grpc-certificate string Local path to the gRPC certificate pem file
  --grpc-key string     Local path to the gRPC key pem file

```

```
--force                Update the certificate even if already present
--- Time Elapsed: 3.350424ms ---
```

**Note**

Fabric and multiple IP do not work with `https|token` (`efa certificates device install --ips <ip-addr> certType [ http|token]`).

Use the following command to install the certificates on multiple devices:

```
efa certificates device install --ip 10.139.44.147-148 --certType https
+-----+-----+
| IP Address | Status |
+-----+-----+
| 10.139.44.148 | Success |
+-----+-----+
| 10.139.44.147 | Success |
+-----+-----+
```

Use the following command to install the HTTPS certificates on the devices in fabric fabric1. If the force option is used, it will update the certificates even if already present:

```
efa certificates device install --fabric fabric1 --certType https --force
+-----+-----+
| IP Address | Status |
+-----+-----+
| 10.139.44.148 | Success |
+-----+-----+
| 10.139.44.147 | Success |
+-----+-----+
```

When you enter the force option, certificates on the devices of interest are updated whether they currently exist or not. If you do not enter the force option, the update reverts to only installing certificates on input devices that do not have them.

Example:

```
Certificates on SLX 10.139.44.147 before and after force:
SLX# show crypto ca certificates
Certificate Type: https; Trustpoint: none
certificate:
SHA1 Fingerprint=CA:7D:13:C6:44:05:71:24:6B:BC:D4:C2:75:95:B6:53:AE:74:03:C0
Subject: CN=slx-10.139.44.147.extremenetworks.com
Issuer: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation Intermediate,
CN=EFA Intermediate CA/emailAddress=support@extremenetworks.com
Not Before: Aug 2 13:42:05 2022 GMT
Not After : Aug 2 13:42:05 2024 GMT
syslog CA certificate(Server authentication):
SHA1 Fingerprint=C4:23:B1:A9:6B:DD:45:6C:AA:9B:85:10:63:65:0E:02:77:7D:68:49
Subject: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation Intermediate,
CN=EFA Intermediate CA/emailAddress=support@extremenetworks.com
Issuer: C=US, ST=CA, L=SJ, O=Extreme Networks, OU=Extreme Fabric Automation,
CN=efa.extremenetworks.com/emailAddress=support@extremenetworks.com
Not Before: Sep 2 13:14:01 2022 GMT
Not After : Aug 30 13:14:01 2032 GMT
oauth2 certificate(OAuth2 token signature validation):
SHA1 Fingerprint=57:55:2F:7A:FO:DB:23:CF:37:67:8D:AE:82:35:D8:2D:18:00:17:9E
Subject: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation,
CN=extremenetworks.com
Issuer: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation,
CN=extremenetworks.com
```

```

Not Before: Sep 2 13:26:27 2022 GMT
Not After : Aug 30 13:26:27 2032 GMT

SLX# show crypto ca certificates
Certificate Type: https; Trustpoint: none
certificate:
SHA1 Fingerprint=73:06:CD:84:F3:C9:12:49:70:88:57:4A:A5:97:43:91:6A:BA:98:A1
Subject: CN=slx-10.139.44.147.extremenetworks.com
Issuer: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation Intermediate,
CN=EFA Intermediate CA/emailAddress=support@extremenetworks.com
Not Before: Aug 2 13:44:24 2022 GMT
Not After : Aug 2 13:44:24 2024 GMT
syslog CA certificate(Server authentication):
SHA1 Fingerprint=C4:23:B1:A9:6B:DD:45:6C:AA:9B:85:10:63:65:0E:02:77:7D:68:49
Subject: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation Intermediate,
CN=EFA Intermediate CA/emailAddress=support@extremenetworks.com
Issuer: C=US, ST=CA, L=SJ, O=Extreme Networks, OU=Extreme Fabric Automation,
CN=efa.extremenetworks.com/emailAddress=support@extremenetworks.com
Not Before: Sep 2 13:14:01 2022 GMT
Not After : Aug 30 13:14:01 2032 GMT
oauth2 certificate(OAuth2 token signature validation):
SHA1 Fingerprint=57:55:2F:7A:F0:DB:23:CF:37:67:8D:AE:82:35:D8:2D:18:00:17:9E
Subject: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation,
CN=extremenetworks.com
Issuer: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation,
CN=extremenetworks.com
Not Before: Sep 2 13:26:27 2022 GMT
Not After : Aug 30 13:26:27 2032 GMT

```

XCO Certificates

All of the XCO components produce and use different certificates.

1. **App Server Certificate:** The certificate of XCO server for secure communication with the clients. This certificate is used on port 443 (default XCO), 8078 (monitor service of XCO), and 6514 (syslog listener on XCO).
2. **Intermediate CA Certificate:** Certificate Authority, which is the issuer of client and server certificates of XCO and HTTPS certificate of SLX.
3. **Root CA Certificate:** Certificate Authority, which is the issuer of Intermediate CA certificate.
4. **JWT Certificate:** The RSA public key for JWT verification. This is also used to send user context from XCO to SLX.
5. **K3s Server Certificate (Internal):** XCO uses K3s for management of services. This certificate is for secure communication of k3s with clients
6. **K3s CA Certificate (Internal):** XCO uses K3s for management of services. These certificates are used for generating all the certificates of K3s.
7. **Host Authentication Service Certificate (Internal):** The server certificate of host authentication service on XCO.
8. **Galera Certificate:** XCO uses Mariadb database with galera cluster for replication. This certificate enables SSL for the replication across nodes. This is only applicable for multi-node installation of XCO.

The following tables provide information about XCO certificates.

For Alerts related to Alarms or Notifications, see [Fault Management – Alerts](#).

SSL/TLS Certificate of XCO

Location in TPVM deployment	/apps/efadata/certs/own/tls.crt
Location in server deployment	/opt/efadata/certs/own/tls.crt
Description	The certificate of XCO server for secure communication with the clients. The same certificate is used on port 443 (default XCO), 8078 (monitor service of XCO), 6514 (syslog listener on XCO), 8079 (host authentication service of XCO)
Default Validity Period	Expires in 3 years from installation. Reset after every subinterface creation/upgrade
Impact on the system	If the certificate expires, then the server communication with SSL verification enabled will fail. Disables syslog messages from the devices
Renewal Procedure	Use the efa certificate server renew command as described in the XCO Server Certificate on page 61.
Alarm/Notification	Notification is sent to XCO subscribers from 30 days to expiry and warning message on every login from 7 days to expiry. Notification is sent to XCO subscribers: 1. After 30 days of expiry 2. Expired certs 3. Renewal certs

K3s CA Certificate

Location in TPVM deployment	/apps/rancher/k3s/server/tls/server-ca.crt
Location in server deployment	/var/lib/rancher/k3s/server/tls/server-ca.crt
Description	XCO uses K3s for management of services. These certificates are for secure communication of K3s with clients.
Default Validity Period	Expires in 10 years from the date of installation.
Impact on the system	
Renewal Procedure	K3s CA on page 64.
Alarm/Notification	Notification is sent to XCO subscribers: 1. After 30 days of expiry 2. Expired certs 3. Renewal certs

Intermediate CA Certificate of XCO

Location in TPVM deployment	/apps/efadata/certs/ca/extreme-ca-cert.pem
Location in server deployment	/opt/efadata/certs/ca/extreme-ca-cert.pem

Description	The certificate of Certificate Authority, which is the issuer of client and server certificates of XCO and HTTPS certificate of SLX. Same certificate is seen as SyslogCA on SLX
Default Validity Period	Expires in 10 years from the date of installation
Impact on the system	
Renewal Procedure	XCO Intermediate CA on page 60
Alarm/Notification	Not available Notification is sent to XCO subscribers: 1. After 30 days of expiry 2. Expired certs 3. Renewal certs

Root CA Certificate of XCO

Location in TPVM deployment	/apps/efadata/certs/ca/extreme-ca-root.pem
Location in server deployment	/opt/efadata/certs/ca/extreme-ca-root.pem
Description	The certificate of Certificate Authority, which is the issuer of Intermediate CA certificate
Default Validity Period	Expires in 20 years from the date of installation
Impact on the system	
Renewal Procedure	XCO Root CA on page 59
Alarm/Notification	XCO Certificate Expiry Notice XCO Certificate Expired XCO Certificate Upload or Renewal

HTTPS Certificate of SLX

Location in TPVM deployment	/apps/efadata/certs/slx-<IP>.extremenetworks.com-cert.pem
Location in server deployment	/opt/efadata/certs/slx-<IP>.extremenetworks.com-cert.pem
Description	The certificate of SLX Web Server (Apache) for secure communication with the device from XCO
Default Validity Period	Expires in 2 years from installation
Impact on the system	System will not use encryption for HTTPS requests
Renewal Procedure	HTTPS Certificates on page 51
Alarm/Notification	Notification is sent to XCO subscribers from 30 days of expiry.

GRPC Certificate for OS ONE

Location in TPVM deployment	/apps/efadata/certs/slx-<IP>.extremenetworks.com-cert.pem
Location in server deployment	/opt/efadata/certs/slx-<IP>.extremenetworks.com-cert.pem
Description	The certificate of OS ONE Web Server (Apache) for secure communication with the device from OS ONE
Default Validity Period	Expires in 2 years from installation
Impact on the system	The system will not use encryption for HTTPS requests
Renewal Procedure	HTTPS Certificates on page 51
Alarm/Notification	Notification is sent to XCO subscribers from 30 days before expiry.

K3s Certificate – XCO internal

Location in TPVM deployment	/apps/rancher/k3s/server/tls/
Location in server deployment	/var/lib/rancher/k3s/server/tls/
Description	XCO uses k3s for management of services. This certificate is for secure communication of k3s with clients
Default Validity Period	Expires in 1 year from installation. Reset after every upgrade of XCO
Impact on the system	
Renewal Procedure	K3s Server Certificate on page 65
Alarm/Notification	XCO Certificate Expiry Notice

JWT Signing or Verification – XCO internal

Location in TPVM deployment	/apps/efadata/certs/cert.crt.pem
Location in server deployment	/opt/efadata/certs/cert.crt.pem
Description	The RSA public key for JWT verification. This is also used to send user context from XCO to SLX. The same certificate is seen as Oauth certificate on SLX. It is applicable for both SLX and OS ONE.
Default Validity Period	Expires in 10 years from the date of installation
Impact on the system	Disables login to XCO
Renewal Procedure	JWT Certificate on page 64
Alarm/Notification	XCO Certificate Expiry Notice Managed Device Certificate Expiry Notice Managed Device Certificate Expired XCO Certificate Upload or Renewal Managed Device Certificate Upload or Renewal

Galera Certificate

Location in TPVM deployment	/apps/efadata/galera/galera.pem
Location in server deployment	/opt/efadata/galera/galera.pem
Description	The certificate enables SSL for the replication across the nodes. This is only applicable for multi-node deployment of XCO.
Default Validity Period	Expires in three years from the date of installation which is reset on every upgrade. There is no down time when the certificates are renewed.
Impact on the system	Replication of data between the nodes will fail.
Renewal Procedure	Galera Certificate on page 66
Alarm/Notification	NA

XCO Root CA

XCO is shipped with Root CA that is used to generate Intermediate CA. The Root CA is unique across each XCO and is generated during installation.

Location

- **TPVM:** /apps/efadata/certs/ca/extreme-ca-root.pem
- **Server:** /opt/efadata/certs/own/extreme-ca-root.pem

Expiry and Alerts

The XCO Root CA is valid for 20 years from the date of installation. It supports the following alerts which effects the health of XCO security subsystem:

- CertificateExpiryNoticeAlert
- CertificateExpiredAlert
- CertificateUnreadableAlert

For more information, see [Fault Management – Alerts](#).

Renewal

You can renew or regenerate the root CA by using either script or command.

To renew or regenerate the Root CA, run the renewal script **efa_renew_certs.sh**.

```
sudo bash <path to the script>/efa_renew_certs.sh --type rootca
```

To renew or regenerate the Root CA, run the **efa certificate server renew** command.

```
efa certificate server renew --cert-type
```

**Note**

In TPVM, the renewal script and command are available in the /apps/efa/ and /opt/efa/ directory of a server.

After the Root CA is updated,

- New Intermediate CA is generated
- New XCO Server Certificate is generated. If a third-party certificate is used, then the server certificate generation is skipped.

On renewal of certificate, a `CertificateRenewalAlert` is raised which changes the health of the system to green.

XCO Intermediate CA

XCO is shipped with Intermediate CA that is used to

1. Generate server certificate of XCO
2. Generate HTTPS certificate of SLX
3. Connect from Syslog server of SLX

During an upgrade, the old certificates are retained, and will not be regenerated.

Location

- **TPVM:** `/apps/efadata/certs/ca/extreme-ca-cert.pem`
- **Server:** `/opt/efadata/certs/own/extreme-ca-cert.pem`

Expiry and Alerts

The XCO Intermediate CA is valid for 10 years from the date of installation. It supports the following alerts which effects the health of XCO security subsystem:

- `CertificateExpiryNoticeAlert`
- `CertificateExpiredAlert`
- `CertificateUnreadableAlert`

For more information, see [Fault Management – Alerts](#).

Renewal

You can renew or regenerate the Intermediate CA by using either script or command.

To renew or regenerate the Intermediate CA, run the renewal script `efa_renew_certs.sh`.

```
sudo bash <path to the script>/efa certificate server renew.sh --type intermediatca
```

To renew or regenerate the Intermediate CA, run the **efa certificate server renew** command.

```
efa certificate server renew --cert-type
```



Note

In TPVM, the renewal script and command are available in the `/apps/efa/` and `/opt/efa/` directory on a server installation.

After the Intermediate CA certificate is updated,

- A new XCO server certificate is generated. If a third-party certificate is used, then the server certificate generation is skipped.
- The Syslog certificates for the registered devices are automatically updated.
- You must manually update the HTTPS certificate on the devices.

For more information about updating the certificates, see [HTTPS Certificates](#) on page 51 for SLX.

On renewal of certificate, `CertificateRenewalAlert` is raised which changes the health of the system to green.

XCO Server Certificate

XCO is shipped with a self-signed certificate that is generated during installation. It is signed by the [XCO Intermediate CA](#) on page 60 certificate. This certificate is used on the following ports:

- **443**: Secure installation of XCO
- **8078**: Monitoring service of XCO
- **6514**: RASlog server on port 6514 to connect with devices

Third-party Certificate

You can replace server certificate with a third-party certificate acquired through trusted CAs (for example, Verisign or GoDaddy). The third-party certificate must be present in the host device that is running XCO. You can then install it with the following command:

```
$ efa certificate server --help

Usage:
  efa certificate server [flags]
  efa certificate server [command]

Available Commands:
  renew          Renew certificates for EFA

Flags:
  --certificate string    Certificate for EFA
  --key string           Key File for the certificate
  --cacert string        CA Certificate File
  --client-key string    Client Key File for the certificate
  --client-cert string   Client Certificate for EFA
```

Example:

```
efa certificate server --certificate=my_server.pem --key=my_server.key --cacert=ca-
chain.pem --client-cert=my_client.pem --client-key=my_client.key

Please wait as the certificates are being installed...
```

```
Certificates were installed!
--- Time Elapsed: 30.946303683s ---
```



Note

- If you install your own server certificate to use with the XCO HTTPS server, be sure to reinstall the certificate when you upgrade XCO
- Generate the third-party certificates and keys without a passphrase. Certificate installation may fail if you generate the third-party certificates and keys with passphrase.
- Ensure that the certificate that is uploaded has validity of at least 90 days.
- XCO relies on common name and the SAN IPs of the certificate. For a single-node deployment, the SAN IP field must have the management IP of the system. In multi-node deployment, ensure that the node IPs and the VIP are present.
- If there are any multiaccess subinterfaces, be sure to add these IPs to the SAN IPs when you generate a certificate.

To upload third-party certificates for HTTPS server on SLX, use the following CLI command. This works only to install certificates on a single device at once.

```
(efa:extreme)extreme@tpvm:/apps/test/certs$ efa certificate device install --ip=10.x.x.x
--cert-type https --https-certificate server.crt --https-key my_server.key

WARNING: This will restart the HTTP service on the devices and services will not be able
to connect till the operation is complete. Do you want to proceed [y/n]?
y
+-----+-----+
| IP Address | Status |
| 10.20.61.171 | Success |
+-----+-----+
--- Time Elapsed: 38.516844258s ---
```

The device must have the new certificates uploaded.

```
slx-171# show crypto ca certificates
Certificate Type: https; Trustpoint: none
certificate:
SHA1 Fingerprint=D8:49:5F:12:AC:FE:BB:CB:95:C2:AC:6B:AF:B6:5B:9E:24:66:59:7D
Subject: CN=10.x.x.x/subjectAltName=IP=10.20.61.171
Issuer: C=US, O=xyz, OU=abcd, CN=INTERIM-CN
Not Before: Feb 10 11:23:36 2022 GMT
Not After : Jun 25 11:23:36 2023 GMT
```

XCO utilizes the third-party certificates for northbound access. Prior to XCO 3.2.0, when you run any upgrade or node-replacement procedure, the third-party certificate is replaced with the default certificates of XCO.

It retains the certificates that you have installed during any deployment activities.

In case of any issues while installing the third-party certificates, it will revert back to use the default certificates that are shipped with XCO. The validity of the third-party certificates is verified during XCO upgrade and initial upload of the third-party certificates.

Location

- Default certificate
 - **TPVM:** /apps/efadata/certs/own/tls.crt
 - **Server:** /opt/efadata/certs/own/tls.crt
- Third-party Certificate
 - **TPVM:** /apps/efadata/certs/thirdparty/tls.crt
 - **Server:** /opt/efadata/certs/thirdparty/tls.crt
- Third-party CA Certificate
 - **TPVM:** /apps/efadata/certs/thirdparty/custom-ca-chain.pem
 - **Server:** /opt/efadata/certs/thirdparty/ custom-ca-chain.pem

Expiry and Alerts

The certificate is valid for 3 years from the date of installation. It is regenerated whenever a new multiaccess subinterface is created or deleted from XCO.

Legacy notification is sent to the user if the certificate is going to expire in 30 days. If you do not renew the certificates within 7 days of expiry, a following warning message is displayed on every login to the XCO CLI.

```
(efa:extreme)extreme@tpvm:/apps/test/certs$ efa login
Password:
Login successful.
Warning: The certificate for 'EFA' will expire on '2022-04-08 14:43:43 +0530 IST'.
--- Time Elapsed: 5.532391719s ---
```

XCO server certificate supports the following alerts which effects the health of XCO security subsystem.

- CertificateExpiryNoticeAlert
- CertificateExpiredAlert
- CertificateUnreadableAlert

For more information, see [Fault Management – Alerts](#).

Renewal

To renew the server certificate, use the following command:

```
(efa:extreme)extreme@tpvm:/apps$ efa certificate server renew
Certificate renewal is successful
--- Time Elapsed: 33.516064167s ---
```



Note

- Renewal is not applicable if the third-party certificates are installed on the system. You must upload a new certificate as described in the "Third-party certificates" section of [HTTPS Certificates](#) on page 51.
- On renewal of certificate or a successful upload, `CertificateRenewalAlert` is raised which changes the health of the system to green.

JWT Certificate

XCO uses JSON Web Tokens for authentication which uses RSA key pair for signing and verification of the tokens.

Location

- **TPVM:** /apps/efadata/certs/cert.crt.pem
- **Server:** /opt/efadata/certs/cert.crt.pem

Expiry and Alerts

The certificate is valid for 10 years from the date of installation. It supports the following alerts which effects the health of XCO security subsystem:

- CertificateExpiryNoticeAlert
- CertificateExpiredAlert
- CertificateUnreadableAlert

For more information, see [Fault Management – Alerts](#).

Renewal

To renew or regenerate token signing certificate, use the following command:

```
(efa:extreme)extreme@tpvm:/apps$ efa certificate server renew --cert-type=token
Certificate renewal is successful.
--- Time Elapsed: 27.233017418s ---
```

After the token certificate is updated, it has to be pushed to all the registered devices. For more information about updating the certificates, see [OAuth Certificate](#) on page 51 for SLX.

On renewal of the certificate, `CertificateRenewalAlert` is raised which changes the health of the system to green.

K3s CA

XCO uses K3s for management of microservices which comes up with its own certificates.

Location

- **TPVM:** /apps/rancher/k3s/server/tls/server-ca.crt
- **Server:** /var/lib/rancher/k3s/server/tls/server-ca.crt

Expiry and Alerts

The certificate is valid for 10 years from the date of installation and is regenerated after every upgrade. It supports the following alerts which effects the health of XCO security subsystem:

- CertificateExpiryNoticeAlert
- CertificateExpiredAlert
- CertificateUnreadableAlert

For more information, see [Fault Management – Alerts](#).

Renewal

You can renew or regenerate the K3s CA by using either script or command.

To renew or regenerate the K3S CA, use the renewal script **efa_k3s_renew_certs.sh**.

```
sudo bash <path to the script>/efa_k3s_renew_certs.sh --type ca
```

To renew or regenerate the K3S CA, use the **efa certificate server renew** command.

```
efa certificate server renew --cert-type
```



Note

- In TPVM, the renewal script and command are available in /apps/efa/ and /opt/efa/ on a server installation.
- If there are any third-party certificates already installed on XCO reinstall these certificates after K3s CA certificates are regenerated.

On renewal of the certificate, `CertificateRenewalAlert` is raised which changes the health of the system to green.

K3s Server Certificate

XCO uses K3s for management of microservices which comes up with its own certificates.

Location

- **TPVM:** /apps/efadata/certs/ca/extreme-ca-cert.pem
- **Server:** /opt/efadata/certs/own/extreme-ca-cert.pem

Expiry and Alerts

The certificate is valid for one year from the date of installation which is reset on every upgrade. It supports the following alerts which effects the health of XCO security subsystem:

- CertificateExpiryNoticeAlert
- CertificateExpiredAlert
- CertificateUnreadableAlert

For more information, see [Fault Management – Alerts](#).

Renewal

You can renew or regenerate the K3s CA by using either script or command.

You can perform the renewal of K3s Server certificate only when:

- K3s server certificate has expired
- K3s server certificates expiry is less than 90 days



Note

In TPVM, the renewal script and command are available in the /apps/efa/ and /opt/efa/ directory on a server installation.

To renew or regenerate the K3S server certificate, use the renewal script **efa_k3s_renew_certs.sh**.

```
sudo bash <path to the script>/efa_k3s_renew_certs.sh --type server
```

To renew or regenerate the K3S server certificate, use the **efa certificate server renew** command.

```
efa certificate server renew --cert-type
```

On renewal of the certificate, `CertificateRenewalAlert` is raised which changes the health of the system to green.

Host Authentication Certificate

Before XCO 3.2.0 release, Host Authentication service runs on the XCO host with its own certificate with an expiry of 10 years. This certificate had no renewal procedure.

The Host Authentication service runs on port 8079. The port is closed from external access. This service reuses the XCO server certificate on 443 with an expiry of 3 years.

All the operations that are performed on the server certificate are applied on the Host Authentication service. This includes uploading of third-party certificate, renewal and so on.



Important

From XCO 3.2.0, the Host Authentication Certificate is not present anymore. You can use the XCO server certificate.

Galera Certificate

XCO uses SSL encryption for the communication between the nodes in a multi-node deployment.

XCO uses the MariaDB and Galera services to implement an HA deployment.

- **MariaDB and Galera** – The registrations and configurations for devices, fabrics, and tenants in XCO are stored in a group of databases that are managed by MariaDB. Two nodes operate on a single MariaDB server and utilize Galera clustering technology to synchronize the business state in between the two nodes during standard operation.

When deploying the XCO cluster with multiple nodes, Galera components are automatically configured to communicate over SSL. This SSL configuration does not affect the communication between the cluster servers and their clients. During installation, the SSL configuration generates certificates for the Galera servers. These certificates are signed by the XCO Intermediate CA certificate and remain valid for three years from the date of installation. Upgrades reset the certificate validity period, and there's no downtime when renewing these certificates.

Location

- **TPVM:** /apps/efadata/galera/galera.pem
- **Server:** /opt/efadata/galera/galera.pem

Renewal

To renew Galera certificates, use the `efa certificate server renew` command.

For information about commands and supported parameters to renew Galera certificates, see [.ExtremeCloud Orchestrator Command Reference, 4.1.0](#)

External Certificates

The following tables provide information about external certificates.

Certificate	Location in TPVM deployment	Location in server deployment	Description
Syslog CA (Notification subscriber)	Notification service database in mariadb	Notification service database in mariadb	Connect to external Syslog server for sending notifications over RELP.
Webhook CA	Notification service database in mariadb	Notification service database in mariadb	Connect to external Syslog server for sending notifications over webhooks.
LDAP CA	Auth service database in mariadb	Auth service database in mariadb	Connect to external LDAP server

Alarms for Auto Certificate Renewal Failure

XCO 3.8.0 introduces alerts and alarms for certificate renewal failures in addition to the existing certificate expiry notifications and certificate expired alerts and alarms. The new functionality doesn't impact existing certificate expiry alarms and alerts.

XCO currently has basic certificate management with manual enrollment and renewal, certificate validation, and expiration alarms. This feature expands on that by adding automated certificate renewal and integration with external Certificate Authorities (CAs).



Note

Renewal failures will trigger alerts and alarms, provided the system is functioning properly.

- Automatic Renewal Failure Alerts: New alarms when automatic renewal of internal certificates fails. This excludes automatic renewals during XCO upgrades, as the system won't be active.
- Manual Renewal of XCO CA Certificates: A new alert to capture renewal failures for specific certificates, including:
 - App Server Certificate
 - Default Intermediate CA
 - Default Root CA
 - WT Certificate
 - K3s Server Certificate

- K3s CA
- Galera Certificate

The following table describes the existing renewal mode for various certificate types:

Certificate Name	Validity	Existing Renewal Mode	CLI Command for Manual Renewal
App Server Certificate	3 yrs	Automatic Renewal during XCO upgrade and Manual	efa certificate server renew –cert-type server
Default Intermediate CA	10 yrs	Manual Renewal	efa certificate server renew –cert-type intermediate-ca
Default Root CA	20 yrs	Manual Renewal	efa certificate server renew –cert-type root-ca
JWT Certificate	10 yrs	Manual Renewal	efa certificate server renew –cert-type token
K3s Server Certificate	1 yrs	Automatic Renewal during XCO upgrade and manual	efa certificate server renew –cert-type k3s-server
K3s CA	10 yrs	Automatic Renewal during XCO upgrade and manual	efa certificate server renew –cert-type k3s-ca
Galera Certificate	3 yrs	Automatic Renewal during XCO upgrade and manual	efa certificate server renew –cert-type galera

Alert Details

```
{
  "Name": "CertificateRenewalFailureAlert",
  "AlertID": 31009,
  "Severity": "Warning",
  "Resource": "/App/System/Security/Certificate?type=",
  "DisplayName": "Certificate renewal failure alert"
}
```

Alarm Details

```
{
  "Name": "CertificateRenewal",
  "AlarmID": 32002,
  "Type": "Security",
  "Severity": "Warning, Critical",
  "Resource": "/App/System/Security/Certificate?device_ip=*&type=",
  "WillClear": true,
  "Description": "Notify when a certificate renewal is failed.",
  "RaiseCondition": [
    {
      "name": "CertificateRenewal_Raise",
      "desc": "Certificate Renewal has failed",
      "salience": 0,
      "when": "Alert.AlertID == 31009",
    }
  ]
}
```

```

        "then": [
            "Alert.Log(CertificateRenewal _Raise')",
            "Alert.RaiseAlarm()",
            "Retract('CertificateRenewal _Raise') "
        ]
    },
    "ClearCondition": [
        {
            "name": "CertificateRenewal_Clear",
            "desc": "Clear the CertificateRenewalFailure alarm when an application
certificate has been renewed.",
            "saliency": 10,
            "when": "Alert.AlertID == 31004",
            "then": [
                "Alert.Log('CertificateRenewal_Clear')",
                "Alert.ClearAlarm()",
                "Retract('CertificateRenewal_Clear') "
            ]
        }
    ]
}

```

Certificate Management Using CMPv2

Learn about how to configure and manage certificates in XCO using CMPv2 (Certificate Management Protocol v2).

The CMPv2-based automated certificate lifecycle management feature introduced in XCO 4.1.0. The feature enables XCO to enroll and automatically renew northbound certificates through an external Certificate Authority (CA), reducing manual administrative overhead and the operational risk associated with certificate expiry.

XCO acts as a CMPv2 client, communicating with a customer-provided external CA (EJBCA) to request, install, and renew certificates for its northbound interfaces. Support for mutual TLS (mTLS) on RELP-based syslog connections is also included.

Use this feature to:

- Reduce manual certificate handling
- Improve operational reliability
- Maintain secure communication across services

Platforms Supported

- XCO 4.1.0
- Single-node and HA deployments

Deliverables

- CMPv2 certificate enrollment (Initial Request) and renewal (Key Update Request), supported on fresh deployments and on all supported upgrade paths from earlier XCO releases.
- CLI-based certificate lifecycle management, including apply, renew, status, and cleanup operations.

- RELP syslog integration with mutual TLS (mTLS) using CMPv2-provisioned client certificates.

Limitations

- Applying updated certificates may require a service restart or configuration reload to take effect.
- CMPv2 cannot be configured during installation. Enable it only after the system is fully operational.
- Only one external CA is supported per system.
- CMPv2 configuration is CLI-only in this release. UI workflows are not supported.
- CMPv2 and third-party certificate (TPC) management are mutually exclusive. Only one mechanism can be active at a time.
- CMPv2 certificates are not included in the XCO backup archives. A restore operation preserves the certificate management state active at the time of backup.
- CMPv2 certificates and related configuration are not included in XCO backups. When you restore an XCO backup, any existing CMPv2 configuration on the target system is retained. After the restore operation completes, you must reconfigure CMPv2. For best results, perform the restore on a newly deployed system.
- Before reverting to a different certificate management mechanism, run **efa certificate cmpv2 clean** to remove the active CMPv2 configuration.

Resilience and Redundancy

If certificate renewal fails, the existing certificate remains active and service continuity is maintained. An alert is generated on failure, and manual remediation is available through the CLI.

Backup and restore operations preserve the existing certificate and its associated configuration. If the system's IP address, VIP, or OVA IP changes, certificate regeneration may be required to reflect the updated identity.



Note

CMPv2 renewal must be initiated before certificate expiry. If a certificate has already expired, automatic renewal is not supported. To recover, reset the end-entity status to NEW on the external CA, then re-enroll using **efa certificate cmpv2 apply**.

To deregister CMPv2, run **efa certificate cmpv2 clean**. This removes all CMPv2 certificate artifacts and returns the system to a state where it can be reconfigured with XCO self-signed or third-party certificates.

Event Log Messages

CMPv2 enrollment, renewal, and failure events are captured in the following logs:

- Installer logs
- Monitor service logs

Include both sources when collecting diagnostic data for support cases.

Licensing

- EJBCA Community Edition is supported by default.
- EJBCA Enterprise Edition features, if required, must be licensed and provided by the customer.
- This feature introduces no additional XCO licensing requirements.

Performance and Scalability

- Renewal checks run once every 24 hours.
- Certificate validation is performed locally; no external validation calls are made during normal operation.
- CPU and memory impact is minimal.
- Expected service interruption during certificate enrollment or renewal: approximately 1.5–2.5 minutes.

Third-Party and Open-Source Components

This feature uses the following third-party and open-source components:

Component	Version	Purpose	License
OpenSSL	v3.5 or later	CMPv2 client implementation for certificate enrollment and renewal (RFC 4210)	Apache License v2.0
EJBCA	Community Edition	External Certificate Authority for CMPv2-based certificate issuance	LGPL v2.1 (Community Edition)

Licensing and distribution

- No third-party or open-source components are redistributed outside the XCO software package.
- This feature introduces no per-device royalties or usage-based licensing obligations.
- Customers are responsible for procuring and managing any licenses required for external CA products, including EJBCA Enterprise Edition.

EJBCA CMPv2 Pre-Configuration Requirements

Learn about the mandatory pre-configuration requirements for the external Certificate Authority (EJBCA) that must be completed before enabling CMPv2 in XCO.

Requirement	Description
EJBCA instance deployed	Deploy an external EJBCA instance and verify it is reachable from XCO over the required network path.
CMP enabled	Enable CMP protocol support in the EJBCA server configuration.

Requirement	Description
CMP alias configured	Create a CMP alias in EJBCA and configure it for client-mode operation.
End-entity and certificate profiles	Before initiating CMP enrollment, configure the required End Entity and Certificate Profiles on the EJBCA server. Enable the <code>Clear Text Password Storage</code> option in the End Entity Profile to allow HMAC-based authentication. This setting enables EJBCA to access the enrollment password and use it to calculate the cryptographic Message Authentication Code (MAC) for CMP requests.
HMAC credentials provisioned	Set up HMAC-based authentication credentials (username and password) on EJBCA for use with the <code>--username</code> and <code>--password</code> parameters.
CA trust chain available	Download the CA trust chain certificate from EJBCA and place it on the XCO host before running the <code>apply</code> command.
RELP mTLS: multiple certificates per entity	If RELP mTLS is required, configure the EJBCA end-entity profile to allow issuance of more than one certificate per entity. A profile restricted to a single certificate prevents the RELP client certificate from being provisioned.
RELP server trust	Install the CA chain (the same file passed to <code>--cacert</code> during <code>apply</code>) on the external RELP syslog server so it can validate XCO client certificates.

Configure CMPv2

You can configure CMPv2.

Before you configure CMPv2, ensure the following:

- XCO 4.1.0 is installed and fully operational.
- An external EJBCA instance is deployed and reachable from XCO.
- CMP is enabled on the EJBCA server.
- A CMP alias is created and configured for client-mode operation.
- End-entity and certificate profiles for CMP enrollment are pre-configured on the EJBCA server.
- HMAC authentication credentials are provisioned on the external CA.
- The CA trust chain certificate is available locally on the XCO host.

All CMPv2 certificate management operations are performed using the **efa certificate cmpv2** command set. Follow this procedure to configure CMPv2.

1. Apply CMPv2 Certificate Management.

The command registers an external CMPv2-compatible CA, enrolls certificates for XCO services, and optionally configures auto-renewal. This is the primary command for enabling CMPv2 certificate management.



Important

This is a disruptive operation. XCO services are unavailable for approximately 1.5–2.5 minutes while certificates are enrolled and applied. Do not run this command during a maintenance-sensitive period without prior planning.

```
efa certificate cmpv2 apply \
  --url <CMP-endpoint-URL> \
  --cacert <local-path-to-CA-chain> \
  [--cert <local-path> --key <local-path>] \
  [--username <string> --password <string>] \
  [--auto-renew enable|disable] \
  [--preferred-renewal-daytime <HH:MM:SS>]
```

• Example 1 — Authentication with interactive password prompt (recommended):

```
$ efa certificate cmpv2 apply \
  --url https://10.37.12.53:8443/ejbca/publicweb/cmp/cmp-alias \
  --cacert MySubCA2-chain.pem \
  --username admin

Enter CMPv2 password:

CMPv2 certificate apply successful.
--- Time Elapsed: 2m25.041s ---
```

• Example 2 — Authentication with auto-renewal enabled:

```
$ efa certificate cmpv2 apply \
  --cacert ManagementCA.crt \
  --username admin \
  --password <password> \
  --url https://10.37.11.38:8443/ejbca/publicweb/cmp/cmp-alias \
  --auto-renew enable

CMPv2 certificate apply successful.
--- Time Elapsed: 2m25.041s ---
```



Caution

Passing `--password` on the command line may expose credentials in shell history. Use the interactive prompt (omit `--password`) when possible.

2. View CMPv2 status.

Displays the current CMPv2 configuration and the operational status of certificate enrollment and renewal.

```
$ efa certificate cmpv2 status
```

Example Output — Certificate Installing/Installed/Renewing/Renewed

```
$ efa certificate cmpv2 status
+-----+-----+-----+-----+-----+-----+
| Name |      URL      | Username | Status | Updated At | Auto Renew |
+-----+-----+-----+-----+-----+-----+
|      |               |          |        |             |            |
```

```

+-----+-----+-----+-----+-----+-----+-----+
+
| test | <CMP-Alias_URL> | Admin | Installing | 2026-04-10 09:27:09.83 | true
|
+-----+-----+-----+-----+-----+-----+
CMPv2 Certificate status detail
--- Time Elapsed: 40.6526ms ---

$ efa certificate cmpv2 status
+-----+-----+-----+-----+-----+-----+-----+
| Name | URL | Username | Status | Updated At | Auto Renew |
|
+-----+-----+-----+-----+-----+-----+
+
| test | <CMP-Alias_URL> | Admin | Installed | 2026-04-10 09:27:09.83 | true
|
+-----+-----+-----+-----+-----+-----+
CMPv2 Certificate status detail
--- Time Elapsed: 40.6526ms ---

$ efa certificate cmpv2 status
+-----+-----+-----+-----+-----+-----+-----+
| Name | URL | Username | Status | Updated At | Auto
Renew |
+-----+-----+-----+-----+-----+-----+
+
| test | <CMP-Alias_URL> | Admin | Renewing | 2026-04-10 09:27:09.83 | true
|
+-----+-----+-----+-----+-----+-----+
CMPv2 Certificate status detail
--- Time Elapsed: 40.6526ms ---

$ efa certificate cmpv2 status
+-----+-----+-----+-----+-----+-----+-----+
| Name | URL | Username | Status | Updated At | Auto
Renew |
+-----+-----+-----+-----+-----+-----+
+
| test | <CMP-Alias_URL> | Admin | Renewed | 2026-04-10 09:27:09.83 | true
|
+-----+-----+-----+-----+-----+-----+
CMPv2 Certificate status detail
--- Time Elapsed: 40.6526ms ---

```

3. Run manual certificate renewal.

Initiates an immediate certificate renewal (KUR) request to the external CA. Use this command when auto-renewal is disabled or when renewal must be performed ahead of schedule.



Caution

Manual renewal causes a brief service disruption of approximately 1.5–2.5 minutes. A confirmation prompt is presented before the operation proceeds.

```
$ efa certificate cmpv2 renew
```

Example Output:

```
$ efa certificate cmpv2 renew
```

```
WARNING: Executing above command may cause temporary downtime or service disruption.
Do you want to proceed? (yes/no): yes
```

```
Successfully renewed CMPv2 certificate.
```

4. Clean Up CMPv2 Configuration.

Removes CMPv2 certificate artifacts and configuration from XCO, returning the system to its previous certificate management state. Run this command before switching to a different certificate management mechanism.

```
$ efa certificate cmpv2 clean
```

Example Output:

```
$ efa certificate cmpv2 clean
```

```
CMPv2 certificate clean successful.
--- Time Elapsed: 2m43.084s ---
```



Note

After running `efa certificate cmpv2 clean`, also revoke or remove the corresponding end entity and certificate profile from the EJBCA server to complete cleanup on the CA side.

5. View Certificate Expiry and Managing Module.

Displays the expiry date and managing module (CMPv2 or XCO Self Signed) for all certificates on the system. Use this command to verify which certificates are actively managed by CMPv2.

```
$ efa certificate expiry show
```

Example Output:

```
$ efa certificate expiry show
```

Name	Status	Managing Module	Expiry
App Server Certificate	Valid	CMPv2	2026-05-05T09:37:45Z
Default Intermediate CA	Valid	XCO Self Signed	2036-05-01T09:37:00Z
Default Root CA	Valid	XCO Self Signed	2046-04-29T09:36:59Z
JWT Certificate	Valid	XCO Self Signed	2036-05-01T09:40:53Z
K3s Server Certificate	Valid	XCO Self Signed	2027-05-04T09:18:01Z
K3s CA	Valid	XCO Self Signed	2036-05-01T09:18:01Z
RELPL Syslog client certificate	Valid	CMPv2	2026-05-05T09:37:46Z
Galera Certificate	Valid	XCO Self Signed	2036-04-07T07:09:55Z

```
Certificate Expiry List detail show
--- Time Elapsed: 224.1ms ---
```



Note

Certificates showing 'CMPv2' in the Managing Module column are enrolled and renewed automatically. Certificates showing 'XCO Self Signed' continue to be managed by XCO internally and are not affected by CMPv2 operations.

Auto Renewal of Certificate

Enabled by default and triggered before expiration.

When auto-renewal is enabled, XCO evaluates certificate validity once every 24 hours. Renewal is triggered approximately seven days before the certificate expiry date. The renewal process

runs automatically in the background and does not require manual intervention under normal conditions.

Behaviour	Description
Renewal check frequency	Once every 24 hours.
Renewal window	Initiated approximately seven days before certificate expiry.
Failure handling	Existing certificates remain active if renewal fails. Alerts are generated and manual remediation is possible via CLI.
Expired certificate	Auto-renewal is not supported for already-expired certificates. See Section 5.1 for recovery steps.
IP or VIP address changes	Certificate regeneration may be required after IP address, VIP, or OVA IP changes to ensure the updated identity is reflected.

Switch Certificate Management Mechanism

You can switch from CMPv2 to TPC or XCO self-signed certificates.

CMPv2 and third-party certificate (TPC) management are mutually exclusive.

Follow this procedure to switch from CMPv2 to TPC or XCO self-signed certificates. The transition requires explicit user confirmation and does not affect checkpointed system state.

1. Remove CMPv2 configuration.

```
$ efa certificate cmpv2 clean
```

2. Apply the target certificate mechanism.

- a. To revert to XCO self-signed certificates, run the following command:

```
$ sudo mv $EFA_DATA_DIR/certs/cmpv2 /tmp/cmpv2_bkp
$ k3s kubectl delete secret efasecret-tls -n kube-system
$ efa certificate server renew --cert-type=server
```

- b. To apply third-party certificates, run the following command:

```
$ sudo mv $EFA_DATA_DIR/certs/cmpv2 /tmp/cmpv2_bkp
$ k3s kubectl delete secret efasecret-tls -n kube-system
$ efa certificate server --key=</path/to/key> \
  --certificate=</path/to/cert> \
  --cacert=</path/to/ca>
```



Note

If third-party certificates were previously installed, replace the first step with:
`sudo mv $EFA_DATA_DIR/certs/thirdparty /tmp/thirdparty_bkp.`

Compatibility and Upgrade Considerations

Backward Compatibility

This feature does not introduce changes to existing checkpointed data structures and does not modify ISSU or rolling upgrade workflows. Backward compatibility is fully preserved between XCO 4.0.2 and XCO 4.1.0.

Certificate Mechanism Exclusivity

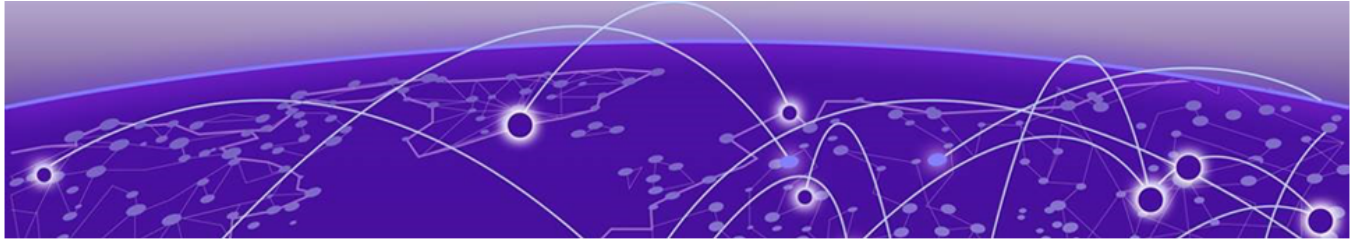
- CMPv2 and third-party certificate (TPC) management are mutually exclusive. Only one mechanism can be active at a time.
- CMPv2 takes precedence when configured.
- Transitioning between mechanisms requires running `efa certificate cmpv2 clean` before applying an alternative configuration.

Upgrade Path

CMPv2 certificate management is supported on fresh deployments and on supported upgrade paths from previous XCO releases. After upgrade, existing certificates remain active. CMPv2 must be explicitly enabled post-upgrade — it is not activated automatically.

Backup and Restore

CMPv2 certificates are not included in XCO backup archives. After a restore operation, the system preserves the certificate management state that was active prior to the restore. Re-enrollment through `efa certificate cmpv2 apply` may be required if the restored state does not include a valid CMPv2 configuration.



Security Hardening

[Overview of System Hardening in XCO](#) on page 78
[System Hardening for CIS-CAT Assessments](#) on page 79
[Hardening Script for Ubuntu Linux Host Servers](#) on page 84
[Security Hardening for SLX-OS in XCO](#) on page 84
[The iptables Policy](#) on page 91
[Secure the Grub Boot Loader](#) on page 93
[Install the Linux Audit System](#) on page 94
[Secure NTP and XCO](#) on page 95
[Secure DNS and XCO](#) on page 96
[Detect Rootkits with rkhunter](#) on page 97
[Configure Remote Syslog \(RELP\) Securely](#) on page 98
[Configuration Files Modification Guidelines](#) on page 100

Learn how to install and use open source security tools to achieve an enhanced security stance for XCO.

Overview of System Hardening in XCO

Get an overview of techniques for hardening security in XCO.

Learn about security hardening guidance for ExtremeCloud Orchestrator (XCO), with an emphasis on the installation and usage of open source security tools to achieve a hardened operational security stance. It is assumed that you have some basic knowledge of security principles and operations of the Linux operating system and associated technologies.



Note

This document assumes that XCO has been installed in secure mode and is operational. For details on how to achieve this, refer to the [ExtremeCloud Orchestrator CLI Administration Guide, 4.1.0](#), which includes details on XCO security options and commands.

**Note**

After upgrading SLX-OS, inventory device updates may intermittently fail with 401 Unauthorized errors. The root cause is linked to a token validation issue triggered by service restarts during hardening script execution. These service restarts initiate HA events, which invalidate tokens. Manual updates succeed, but REST API calls fail due to 401 Unauthorized responses.

Log in to XCO (**efa login**) only after all dependent components have completed deployment and upgrade.

- Do not log in to XCO (via CLI or REST) during deployment.
- Avoid obtaining an `access_token` or starting an XCO session before all components are finished deploying.
- Wait until the following are fully completed:
 - XCO deployment or upgrade
 - SLX-OS upgrade
 - TPVM upgrade
 - Hardening script (HS) execution and completion

The following security hardening topics are included in this document:

- **CIS-CAT security hardening:** Details of a custom python script from Extreme Networks that hardens the underlying operating system.
- **Iptables firewall:** Securing the XCO networking stance.
- **Grub boot loader security:** How to set a hardened security posture for Grub.
- **System auditing with auditd:** Instructions for monitoring various aspects of system runtime activities.
- **OSSEC HIDS installation and usage:** A broad set of indicators relevant for host intrusion detection.
- **Authenticated NTP:** How to ensure that NTP communications are authenticated.
- **Secure DNS:** Details about encrypted DNS communications.
- **Detecting rootkits with rkhunter:** Specialized run time checks for various types of Linux rootkits.

System Hardening for CIS-CAT Assessments

TPVM provides a security hardening capability in the form of a python script located at `/opt/security/extr-granite.py` (for TPVM installations valid from version 4.5.0). The goal of this script is to modify various system security settings to achieve a more secure state under the examination of the CIS-CAT host scanner. Specifically, XCO running on TPVM is deployed into the Ubuntu server environment, and it is this environment that is hardened by `extr-granite.py`.

**Note**

The `/opt/security/extr-granite.py` script should not be exported to other 3rd party systems.

To perform the steps that follow, make sure you have Java Runtime Environment (JRE) installed. Also, ensure that the CIS-CAT scanner has been copied to the TPVM at `/root/cis-cat/Assessor-CLI`.

**Note**

CIS-CAT scanner is not bundled with TPVM. You must procure an external CIS-CAT license and install it on the TPVM environment.

The procedure itself involves the running of the `extr-granite.py` script. Notable aspects of the script are:

- The `extr-granite.py` script keeps a dedicated Git repository for all changes it makes to the underlying file system. This repository is located at `/opt/extr-granite-hardening/OS-files-git`.
- Git tags are used by `extr-granite.py` for every run cycle. This allows easy comparison of what the script changes on the host operating system from one run to the next.
- A comprehensive log file is kept at `/opt/extr-granite-hardening/hardening-log`.

The hardening script can be run multiple times, and even run at every boot using the init scripts. Every run receives its own Git tag in the `OS-files-git` repository. This makes it possible to track the changes the script has made, going back to the initial import.

It is discouraged to add, remove, or modify the hardening script audit rules unless necessary.

Hardening script must be run Post XCO installation for the audit logs feature to be function properly.

The Auditd default rules are set by the hardening script (`extr-granite.py`), which includes the necessary login/logout related rules. If the hardening script is not run, the Host OS login/logout functionality may not function accordingly. These rules should not be removed. It is the customer's responsibility to ensure Auditd rules are maintained. Editing the Auditd rules is not part of the XCO functionality. The hardening script sets the Auditd rules according to industry well-known CIS-CAT recommendations.

Perform the following steps to run the security script and verify that it is working:

1. Install CIS-CAT on the host and produce a scan result.

Here is an example of the abbreviated output. There is a CIS-CAT score of 56.11% on the TPVM.

```
-----  
***** Assessment Results Summary *****  
-----
```

```
Total # of Results: 241  
Total Scored Results: 180  
Total Pass: 101 Total  
Fail: 78  
Total Error: 1  
Total Unknown: 0  
Total Not Applicable: 0  
Total Not Checked: 20  
Total Not Selected: 37  
Total Informational: 4
```

```
-----  
***** Assessment Scoring *****
```



```

-----
Score Earned: 101.0
Maximum Available: 180.0
Total: 56.11%
-----

```

2. Run /opt/security/extr-granite.py.

Here is an example of an abbreviated output.

```

root@node-1:/opt/security# ./extr-granite.py
Initialized empty Git repository in /opt/extr-granite-hardening/OS-files-git/.git/
[master (root-commit) 1e2796b] initial import
1 file changed, 1 insertion(+)
create mode 100644 README
[+] ./extr-granite.py version: 0.89 - Initialized
sending incremental file list
/usr/
/usr/sbin/
/usr/sbin/grub-mkconfig

sent 3,467 bytes received 47 bytes 7,028.00 bytes/sec
total size is 8,219 speedup is 2.34
[master 8b2b4bd] initial import: /usr/sbin/grub-mkconfig, for CIS-CAT test: 1.4.1
Ensure permissions on bootloader config are not overridden
1 file changed, 311 insertions(+)
create mode 100755 usr/sbin/grub-mkconfig
sending incremental file list
/etc/
/etc/sysctl.conf

sent 1,163 bytes received 39 bytes 2,404.00 bytes/sec
total size is 2,683 speedup is 2.23
[master 2f4ad6c] initial import: /etc/sysctl.conf, for CIS-CAT test: 1.5.2 Ensure
address space layout randomization (ASLR) is enabled
1 file changed, 77 insertions(+)
create mode 100644 etc/sysctl.conf
kernel.randomize_va_space = 2
sending incremental file list
/etc/security/
/etc/security/limits.conf

.....
.....
.....

sent 1,168 bytes received 44 bytes 2,424.00 bytes/sec
total size is 2,306 speedup is 1.90
[master 7ef96f4] file: /etc/pam.d/su, CIS-CAT test: 5.7 Ensure access to the su
command is restricted
1 file changed, 1 insertion(+)
sending incremental file list
/etc/group

sent 544 bytes received 36 bytes 1,160.00 bytes/sec
total size is 832 speedup is 1.43
[master 374f6c9] file: /etc/group, CIS-CAT test: 5.7 Ensure access to the su command
is restricted (2)
1 file changed, 1 insertion(+), 1 deletion(-)
sending incremental file list
/etc/profile

sent 436 bytes received 36 bytes 944.00 bytes/sec
total size is 619 speedup is 1.31
[master b305cfe] file: /etc/profile, CIS-CAT test: 5.5.5 Ensure default user shell
timeout is 900 seconds or less

```

```
1 file changed, 3 insertions(+)

[+] Total checks run: 55

root@node-1:/opt/security#
```

The following example shows all changes from the initial import to the tag of the first run:

```
extr-granite-run1
```

Further, the example does the same `git diff`, except it shows just the changes that were made to the original `/etc/ssh/sshd_config` file.

```
root@tpvm:/opt/extr-granite-hardening/OS-files-git# git tag -l
extr-granite-initial-import
extr-granite-run1
root@tpvm:/opt/extr-granite-hardening/OS-files-git# git diff extr-granite-initial-
import extr-granite-run1
diff --git a/etc/issue b/etc/issue
index 80ae21e..5192c40 100644
--- a/etc/issue
+++ b/etc/issue
@@ -1,2 +1 @@
-Ubuntu XX.XX
-
+Extreme Networks, Inc. EFA product. Authorized users only. All activity may be
+monitored and reported.
diff --git a/etc/issue.net b/etc/issue.net
index 5e9e2fa..5192c40 100644
--- a/etc/issue.net
+++ b/etc/issue.net
@@ -1 +1 @@
-Ubuntu XX.XX
+Extreme Networks, Inc. EFA product. Authorized users only. All activity may be
+monitored and reported.
diff --git a/etc/modprobe.d/cramfs.conf b/etc/modprobe.d/cramfs.conf
new file mode 100644
index 0000000..b77c93a
--- /dev/null
+++ b/etc/modprobe.d/cramfs.conf
@@ -0,0 +1 @@
+install cramfs /bin/true
diff --git a/etc/modprobe.d/freevxfs.conf b/etc/modprobe.d/freevxfs.conf
new file mode 100644
index 0000000..72d4aec
--- /dev/null
+++ b/etc/modprobe.d/freevxfs.conf
@@ -0,0 +1 @@
+install freevxfs /bin/true
root@tpvm:/opt/extr-granite-hardening/OS-files-git# ls etc/ssh/sshd_config
etc/ssh/sshd_config
root@tpvm:/opt/extr-granite-hardening/OS-files-git# git help diff^C
root@tpvm:/opt/extr-granite-hardening/OS-files-git# git diff extr-granite-initial-
import extr-granite-run1 ./etc/ssh/sshd_config
diff --git a/etc/ssh/sshd_config b/etc/ssh/sshd_config
index 3f0e52e..f640120 100644
--- a/etc/ssh/sshd_config
+++ b/etc/ssh/sshd_config
@@ -121,4 +121,11 @@ Subsystem sftp      /usr/lib/openssh/sftp-server
#
#       PermitTTY no
#       ForceCommand cvs server
PasswordAuthentication yes
-MaxStartups 30:30:100
```

```
+MaxStartups 10:30:60
+MaxAuthTries 4
+MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512,hmac-
sha2-256
+KexAlgorithms curve25519-sha256,curve25519-sha256@libssh.org,diffie-hellman-
group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,ecdh-sha2-
nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group-exchange-sha256
+LoginGraceTime 60
+Banner /etc/issue.net
```

**Note**

After running the security hardening script, reconnect to the existing TPVM SSH sessions (active/standby/vip) for the new ssh parameters to take effect.

3. Rerun the CIS-CAT auditor and verify that the scan results produce a score greater than 80%.

Note that the hardening results are against version 4.56.0 of the CIS-CAT Assessor.

```
-----
***** Assessment Results Summary *****
-----
```

```

Total # of Results: 300
Total Scored Results: 281
      Total Pass: 229
      Total Fail: 52
      Total Error: 0
      Total Unknown: 0
Total Not Applicable: 0
      Total Not Checked: 14
      Total Not Selected: 0
      Total Informational: 5
-----
```

```
***** Assessment Scoring *****
-----
```

```

      Score Earned: 229.0
      Maximum Available: 281.0
              Total: 81.49%
-----
```

Hardening Script for Ubuntu Linux Host Servers

Hardening Script for Ubuntu Linux Host Servers

Note that the hardening results are against version 4.52.0 of the CIS-CAT Assessor.

As well as the `/opt/security/extr-granite.py` script for TPVM deployments, a security hardening script for Ubuntu Linux 20.04 LTS host servers is available at `/opt/efa/security/extr-granite-server.py`. The script depends on three packages, `auditd`, `audispd-plugins`, and `iptables-persistent`. These packages are not shipped with XCO and they and their dependencies must be installed on the XCO server before using the security script.

To run this procedure, complete the following:

```
(efa:user)user@server2:~/efa33$ cd /opt/efa/security/
(efa:user)user@server2:/opt/efa/security$ ls
extr-granite-server.py
(efa:user)user@server2:/opt/efa/security$ sudo su
root@server2:/opt/efa/security# ls
extr-granite-server.py
root@server2:/opt/efa/security# ./extr-granite-server.py
Initialized empty Git repository in /opt/extr-granite-server-hardening/OS-files-git/.git/
[master (root-commit) 136e544] initial import
1 file changed, 1 insertion(+)
create mode 100644 README
[+] ./extr-granite-server.py version: 0.01 - Initialized
sent 433 bytes received 36 bytes 938.00 bytes/sec
total size is 619 speedup is 1.32
[master f09c26c] file: /etc/profile, CIS-CAT test: 5.5.5 Ensure default user shell
timeout is 900 seconds or less
1 file changed, 3 insertions(+)

[+] Total checks run: 71

root@server2:/opt/efa/security# exit
exit
(efa:user)user@server2:/opt/efa/security$
```

Security Hardening for SLX-OS in XCO

Harden your security for SLX-OS devices in ExtremeCloud Orchestrator.

SLX-OS Device Configuration

As part of security hardening of the SLX-OS device, several configurations are supported from XCO. These configurations are applicable only for the SLX-OS versions 20.3.2 and above. Any SSH server settings change need SSHD to be restarted, and hence any client connected via SSH needs to reconnect..

The following configuration are applied on the SLX-OS device during registration in XCO.

1. SSH Server restarts on the device after the SSH configuration is completed.

Setting	Default Value
SSHD MAC Algorithms	hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512, hmac-sha2-256
SSHD Key Exchange Algorithms	curve25519-sha256, curve25519-sha256@libssh.org, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, diffie-hellman-group-exchange-sha256
Cipher	non-cbc

The following SLX-OS command is for the SSH configuration on SLX-OS devices:

```
SLX# config
Entering configuration mode terminal
SLX(config)# ssh server mac hmac-sha2-512-etm@openssh.com,hmac-sha2-256-
etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256,hmac-sha2-512
% Info: Configuration is successful.For this config to take effect immediately,
restart SSH server via exec command ssh-server restart or save the config and reload.
SLX(config)# ssh server key-
exchange curve25519-sha256,curve25519-sha256@libssh.org,diffie-hellman-group-
exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-
hellman-group14-sha256
% Info: Configuration is successful.For this config to take effect immediately,
restart SSH server via exec command ssh-server restart or save the config and reload.
SLX(config)# ssh server cipher
SLX(config)# ssh server cipher non-cbc
% Info: Configuration is successful.For this config to take effect immediately,
restart SSH server via exec command ssh-server restart or save the config and reload.
SLX(config)# exit
SLX# ssh-server restart

Warning: This operation will disconnect all active SSH sessions.

Are you sure you want to restart the SSH server [y/n]? y
SSH server is going down for restart NOW !!
```

2. The following command shows the SSH configuration parameters on SLX-OS :

```
SLX# show ssh server status
SSH Kex Exchange Algorithm: curve25519-sha256,curve25519-sha256@libssh.org,diffie-
hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-
sha512,diffie-hellman-group14-sha256
SSH Server Rekey Volume: 1024
SSH Server Auth Tries: 6
SSH Server Login Timeout: 120
SSH Server Cipher: non-cbc
SSH Server Mac : hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256,hmac-sha2-512
VRF-Name: mgmt-vrf      Status: Enabled
VRF-Name: default-vrf   Status: Enabled
```

3. The following SLX-OS command disables the Telnet server on mgmt-vrf:

```
SLX# config
Entering configuration mode terminal
SLX(config)# telnet server use-vrf mgmt-vrf shutdown
```

4. The following command shows the Telnet configuration on SLX-OS :

```
SLX# show telnet server status
VRF-Name: default-vrf      Status: Enabled
VRF-Name: mgmt-vrf        Status: Disabled
```

5. The following attributes on the SLX-OS devices (applicable for versions above 20.3.1) are applicable for password configuration:

Setting	Default Value
Max Password Age	90
Force Default Password Change	Disabled

The following SLX-OS command configures the password attributes on SLX-OS devices:

```
SLX# config
Entering configuration mode terminal
SLX(config)# password-attributes max-password-age 365
SLX(config)# password-attributes force-default-password-change
```

6. The following command shows the password configuration on SLX-OS :

```
SLX# show running-config password-attributes
password-attributes force-default-password-change
password-attributes max-password-age 365
```

7. The following SLX-OS command configures TLS on SLX-OS devices: (applicable for versions above 20.3.2):

The minimum version of TLS Configuration on the server is set to 1.2.

```
SLX# config
Entering configuration mode terminal
SLX(config)# management-security
SLX(mgmt-security)# ssl-profile server
SLX(mgmt-sec-ssl-profile-server)# tls min-version 1.2
To view the configuration on SLX,
SLX# show running-config management-security ssl-profile server tls
management-security
  ssl-profile server
    tls min-version 1.2
!
```

- The configuration attributes described above are the default values that are available in XCO on installation.
- The settings are 'enabled' by default. On device registration, the settings are applied on SLX-OS based on the supported versions.
- When you update the settings before device registration, the same values are applied on the device.
- If the settings are changed after device registration, you must manually apply the settings on the specific devices.
- On a device update, if there is any deviation, the AppState will be in 'cfg-refresh'.
- When the device is unregistered from XCO, these settings are removed from the device.

Global Device Security Settings

1. The following command displays the security settings that are configured on the SLX-OS devices:

These settings are common across all devices registered on the XCO installation.

```
efa inventory device secure settings show
```

NAME	VALUE
Min-tls-version	1.2
Mac-algorithm	hmac-sha2-512-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512 hmac-sha2-256
Key-exchange-algorithm	curve25519-sha256 curve25519-sha256@libssh.org diffie-hellman-group14-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group-exchange-sha256
Cipher	non-cbc
Telnet	Disable
Max-password-age	365

2. The following command updates a security setting applicable for the SLX-OS devices:

```
efa inventory device secure settings update --min-tls-version 1.2

efa inventory device secure settings update --mac-algorithm hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512,hmac-sha2-256

efa inventory device secure settings
update --key-exchange-algorithm curve25519-sha256,curve25519-sha256@libssh.org,diffie-hellman-group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group-exchange-sha256

efa inventory device secure settings update --telnet enable --cipher non-cbc --max-password-age 365
```



Note

After modifying any device secure settings, the changes are not automatically applied. You must manually enforce these settings on the devices or fabric, as described in "Device Security Settings".

3. The following command resets the security setting to the default value on the SLX-OS devices:

```
efa inventory device secure settings reset --telnet --cipher --max-password-age
```

--min-tls-version	Reset minimum TLS version to the default value
--mac-algorithm	Reset MAC Algorithms to the default values
--key-exchange-algorithm	Reset Key-Exchange Algorithms to the default values
--cipher	Reset Ciphers to the default values
--telnet	Reset telnet to the default value of disabled

```
--max-password-age          Reset the maximum number of days before
password expiry to the default value
--force-default-password-change  Reset force a change in the default password
to the default value
```

4. The following command enables or disables the security settings on the SLX-OS devices:

If you do not want to configure any security hardening settings on the device, disable the secure settings before device registration.

```
$ efa inventory device secure settings disable
Device secure settings have been disabled.
--- Time Elapsed: 57.000421492s ---
```



Note

If you disable the security settings after device registration, there will not be any change done on the device.

Device Security Settings

Apply the security hardening configuration on the device. You can use this command for enabling security hardening on devices that are already registered in XCO or if there is any update in the security settings.

1. The following command applies the security settings on the SLX-OS devices:

```
efa inventory device secure settings apply [ --ip device-ips | --fabric fabric |
--ip device-ip          Specifies a comma-separated range of device IP addresses.
Example: 1.1.1.1-3,1.1.1.2,2.2.2.2.
--fabric fabric          Specifies fabric name.
```

Example:

```
efa inventory device secure settings apply --ip 1.1.1.1-3,2.2.2.2
efa inventory device secure settings apply --fabric fabric1
```

2. The following command shows the current settings on an SLX-OS device:

```
efa inventory device secure settings show [ --ip device-ip |
--ip device-ip          Specifies a device IP address. Example: 1.1.1.1.
```

Example:

```
efa inventory device secure settings show --ip 1.1.1.1
+-----+-----+
|          NAME          | VALUE |
+-----+-----+
| Min-tls-version        | 1.2   |
+-----+-----+
| Mac-algorithm          | hmac-sha2-512-etm@openssh.com |
|                        | hmac-sha2-256-etm@openssh.com |
|                        | hmac-sha2-512                  |
|                        | hmac-sha2-256                  |
+-----+-----+
| Key-exchange-algorithm | curve25519-sha256              |
|                        | curve25519-sha256@libssh.org   |
|                        | diffie-hellman-group14-sha256  |
|                        | diffie-hellman-group16-sha512  |
```


	diffie-hellman-group18-sha512	
	diffie-hellman-group-exchange-sha256	
+-----+	+-----+	+-----+
Cipher	non-cbc	
+-----+	+-----+	+-----+
Telnet	Disable	
+-----+	+-----+	+-----+
Max-password-age	365	
+-----+	+-----+	+-----+

Drift and Reconcile for Security Settings

Drift is calculated by comparing the settings on the device and the global security settings as these settings are the user intended settings that must be available on the system. SSH server restarts whenever applicable.

Table 9: Drift Reconcile & Idempotency support

Identify Drift	Reconcile configuration	Idempotency
Yes	Yes	Yes

```
(efa:extreme)extreme@tpvm:/opt $ efa inventory drift-reconcile detail --uuid 3d073e9c-879f-4db7-9ab9-852c3f669d51
```

+-----+	+-----+	+-----+
	NAME	
+-----+	+-----+	+-----+
	UUID	3d073e9c-879f-4db7-9ab9-852c3f669d51
+-----+	+-----+	+-----+
	Device IP	10.x.x.x
+-----+	+-----+	+-----+
	Status	success
+-----+	+-----+	+-----+
	Execution Reason	manual
+-----+	+-----+	+-----+
	operation	drift-and-reconcile
+-----+	+-----+	+-----+
	Inventory Status	inventory-dr-success
+-----+	+-----+	+-----+
	Is Inventory config Refreshed	true
+-----+	+-----+	+-----+
	Inventory Duration	15.586983384s
+-----+	+-----+	+-----+
	Fabric Status	fabric-dr-success
+-----+	+-----+	+-----+
	Is Fabric config Refreshed	false
+-----+	+-----+	+-----+
	Fabric Duration	121.479233ms
+-----+	+-----+	+-----+
	Policy Status	policy-dr-success
+-----+	+-----+	+-----+
	Is Policy config Refreshed	false
+-----+	+-----+	+-----+
	Policy Duration	88.764104ms
+-----+	+-----+	+-----+
	Tenant Status	tenant-dr-success
+-----+	+-----+	+-----+
	Is Tenant config Refreshed	false
+-----+	+-----+	+-----+
	Tenant Duration	49.042052ms
+-----+	+-----+	+-----+

+-----+-----+		
Device Update Count	2	
+-----+-----+		
Device Update Total Duration	2m34.074986291s	
+-----+-----+		
Maintenance Mode Disable		
Duration		
+-----+-----+		
Start Time	2022-09-19 20:25:47 +0530 IST	
+-----+-----+		
Last Modified	2022-09-19 20:29:16 +0530 IST	
+-----+-----+		
Duration	3m29.931352961s	
+-----+-----+		

Inventory Service Response:
Config Drift: Device Secure Settings

+-----+-----+		
NAME	APP STATE	CHILD CONFIG
+-----+-----+		
Device Secure Settings	cfg-refreshed	Secure Setting Max Password
		Age
+-----+-----+		

Reconcile Status:

+-----+-----+		
CONFIG-TYPE	APP STATE	ERROR-MESSAGE
+-----+-----+		
NtpAuthKey	Not-Attempted	
SnmpHost	Not-Attempted	
MMONReboot	Not-Attempted	
InterfaceConfig	Not-Attempted	
SnmpUser	Not-Attempted	
DeviceTimezone	Not-Attempted	
ThresholdMonitor	Not-Attempted	
SecureSetting	Success	
NtpDisable	Not-Attempted	
SnmpView	Not-Attempted	
SnmpGroup	Not-Attempted	
DeviceSetting	Not-Attempted	
NtpServer	Not-Attempted	
SnmpCommunity	Not-Attempted	
BreakoutInterface	Not-Attempted	
+-----+-----+		

Fabric Service Response:

Policy Service Response:

Tenant service Response:
--- Time Elapsed: 75.311491ms ---

The following table describes scenarios for the device secure settings:

Scenario	Secure Settings	Device Config
Fresh installation of XCO	Enabled (Default)	Applied on device registration
Fresh installation	Disabled	No settings are applied during registration
Upgrade from prior releases. Security hardening configuration is executed on the device with same configuration as the default settings in XCO.	Enabled (Default)	Device update will result in <code>cfg-in-sync</code>
Upgrade from prior releases. Security hardening configuration is executed on the device with different configuration than the default settings in XCO.	Enabled (Default)	Device update will result in <code>cfg-refresh</code>
Upgrade from prior releases. No security hardening configuration is executed on the device.	Enabled (Default)	Device update will result in <code>cfg-refresh</code>
Upgrade from prior releases	Disabled	Device update will result in <code>cfg-in-sync</code>

The iptables Policy

As a core component of XCO, Kubernetes uses iptables to control the network connections between pods (and between nodes), handling many of the networking and port forwarding rules. XCO builds a custom iptables policy to firewall off services (such as the MySQL database) on the XCO management interface. The XCO iptables policy is instantiated by default at installation time and is enabled at boot, through the system service.

To see the status of the XCO iptables policy, in addition to the policy itself, run the following commands:

```
$ ssh -l extreme 192.168.10.109
Password:
extreme@tpvm:~$ sudo su -
root@tpvm:~#
root@tpvm:~# systemctl status efa-iptables
  efa-iptables.service - iptables rules for EFA
    Loaded: loaded (/lib/systemd/system/efa-iptables.service; enabled; vendor preset:
    enabled)
    Active: active (exited) since Fri 2020-10-09 20:48:03 UTC; 1 day 17h ago
    Main PID: 19384 (code=exited, status=0/SUCCESS)
    Tasks: 0 (limit: 4638)
    CGroup: /system.slice/efa-iptables.service

Oct 09 20:48:03 tpvm systemd[1]: Starting iptables rules for EFA...
Oct 09 20:48:03 tpvm systemd[1]: Started iptables rules for EFA.
root@tpvm:~# iptables -v -nL EFA_INPUT
Chain EFA_INPUT (1 references)
 pkts bytes target      prot opt in     out     source        destination
    0      0 DROP      tcp  --  eth0    *        0.0.0.0/0     0.0.0.0/0     multiport
dports 1024:6513,6515:8077,8079:65535 ctstate NEW
    0      0 DROP      udp  --  eth0    *        0.0.0.0/0     0.0.0.0/0     multiport
dports 1024:65535 ctstate NEW
```

With iptables policy active, it should not be possible to connect to the MySQL database on TCP port 3306 on the management interface from an external host. Use Nmap to verify that port 3306 been firewalled off:

```
# nmap -n -p 3306 -sV 192.168.10.109

Starting Nmap 7.60 ( https://nmap.org ) at 2020-10-11 14:42 UTC
Nmap scan report for 192.168.10.109
Host is up (0.0039s latency).

PORT      STATE  SERVICE VERSION
3306/tcp   filtered mysql

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.81 seconds

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.85 seconds
```

Nmap Scan Output from a Remote System on the VIP

```
Host is up (0.23s latency).

Not shown: 64511 filtered ports, 1018 closed ports

PORT      STATE  SERVICE      VERSION
22/tcp    open   ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
80/tcp    open   http         Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
443/tcp   open   ssl/https
514/tcp   open   shell?
6514/tcp  open   ssl/syslog-tls?
8078/tcp  open   ssl/http     Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
```

Nmap Scan Output on the Active Node on a Multi-node Setup

```
Host is up (0.0020s latency).

Not shown: 65515 closed ports

PORT      STATE  SERVICE      VERSION
22/tcp    open   ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
80/tcp    open   http         Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
111/tcp   open   rpcbind      2-4 (RPC #100000)
443/tcp   open   ssl/https
514/tcp   open   shell?
```

```

3306/tcp open  mysql?
4567/tcp open  tram?
6514/tcp open  ssl/syslog-tls?
8078/tcp open  ssl/http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
8079/tcp open  ssl/http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
8080/tcp open  ssl/http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
8088/tcp open  http        Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
8091/tcp open  http        Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
8092/tcp open  http        Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
10250/tcp open  ssl/http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
24007/tcp open  rpcbind
49152/tcp open  rpcbind
49153/tcp open  rpcbind
49154/tcp open  rpcbind

```

Secure the Grub Boot Loader

To add the Grub boot loader to the security posture, perform the following steps.

Securing the Grub boot loader is an important addition to the security posture for the operating system where XCO is deployed. There are two general phases for securing the boot loader:

- Set a password in the Grub configuration to harden against modifications to the Linux kernel boot-time command line.
- Set a password for the 'root' user to protect against attempts to acquire single-user mode at boot.

1. Set a password in the Grub configuration:

- Acquire root and then run the `grub-mkpasswd-pbkdf2` command (full output is shown below).
- Append the password hash and the string `set superusers="root"` to the file `/etc/grub.d/40_custom`.
- Add `--unrestricted` to the `"CLASS="` definition line in `/etc/grub.d/10_linux`.
- Run the command `update-grub`.

```

root@tpvm:~# grub-mkpasswd-pbkdf2
Enter password:
Reenter password:
PBKDF2 hash of your password is
grub.pbkdf2.sha512.10000.72C8CE3112C007A315A94DD7A63B58392DD00653ACAF8795C8528D83967
FA24105B0B53D0092522460532AF05C60EE3E0C7EAC95213E865DF31580A341188ABC.843EF94A9C8EE8
AC1776F5B88261D1B6DE437A70AEABE3C814764596F696EE5F7FDF912E63B4D47AE3E7BB468A6B639F00
051D142698142EF158E6C141CF38B7
root@tpvm:~# cat >> /etc/grub.d/40_custom
set superusers="root"

```

```

password_pbkdf2 root
grub.pbkdf2.sha512.10000.A577D1C8F13C93B82EA5E25E834D5BD88ECB94A5B42F2DABE4FB7A235F3
A25A12E6542CB5DA9620B2E0342FE28A4F066BE1B99F2EFBE8C0688FBE11FDB3138DD.2C7C81C7FA0404
C768DDCE097B3AA8DD08C042B4FDBA089C0837F91B6C8864EE83B19CBC6D4C5C126E76FA20BE93789920
913B12CAC841CA65EA3BAD5921F8D5
root@tpvm:~# <edit /etc/grub.d/10_linux to make the CLASS line look like the
following>
root@tpvm:~# grep CLASS /etc/grub.d/10_linux | head -n 1
CLASS="--class gnu-linux --class gnu --class os --unrestricted"
root@tpvm:~ # update-grub
Sourcing file `/etc/default/grub'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.4.0-48-generic
Found initrd image: /boot/initrd.img-5.4.0-48-generic
Found linux image: /boot/vmlinuz-5.3.0-40-generic
Found initrd image: /boot/initrd.img-5.3.0-40-generic
Found linux image: /boot/vmlinuz-4.15.0-118-generic
Found initrd image: /boot/initrd.img-4.15.0-118-generic
Found linux image: /boot/vmlinuz-4.15.0-88-generic
Found initrd image: /boot/initrd.img-4.15.0-88-generic
done

```

2. Set a password for the 'root' user by running the following commands:

```

root@tpvm:~# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully

```

Install the Linux Audit System

To install the Linux Audit System, perform the following steps.

XCO is based on the Ubuntu operating system, which by default, does not come with auditd (Linux Audit System) installed. Perform the instructions that follow to install it in XCO. During the installation process, auditd is enabled by default via systemctl and starts writing audit trail log data to the file /var/log/audit/audit.log



Note

Linux Audit System is available by default on TPVM 4.5.10 and above.

To install, run the following command:

```
# apt-get install auditd
```

This is an example output:

```

type=DAEMON_START msg=audit(1591152521.117:3494): op=start ver=2.8.2 format=raw
kernel=5.3.0-53-generic auid=4294967295 pid=27162 uid=0 ses=4294967295 subj=unconfined
res=success
type=CONFIG_CHANGE msg=audit(1591152521.153:23): op=set audit_backlog_limit=8192 old=64
auid=4294967295 ses=4294967295 res=1
type=CONFIG_CHANGE msg=audit(1591152521.157:24): op=set audit_failure=1 old=1
auid=4294967295 ses=4294967295 res=1
type=CONFIG_CHANGE msg=audit(1591152521.157:25): op=set audit_backlog_wait_time=0
old=15000 auid=4294967295 ses=4294967295 res=1
type=SERVICE_START msg=audit(1591152521.157:26): pid=1 uid=0 auid=4294967295
ses=4294967295 msg='unit=auditd comm="systemd" exe="/lib/systemd/systemd" hostname=?
addr=? terminal=? res=success'

```

You can produce more interesting data from the `audit.log` file when you authenticate to the XCO host. In this case, the audit trail data for the user 'extreme' authenticating via SSH is displayed:

```
type=USER_LOGIN msg=audit(1591975761.779:39): pid=7894 uid=0 auid=4294967295
ses=4294967295 msg='op=login acct="extreme" exe="/usr/sbin/sshd" hostname=?
addr=192.168.10.12 terminal=sshd res=failed'
type=USER_AUTH msg=audit(1591975763.219:40): pid=7894 uid=0 auid=4294967295
ses=4294967295 msg='op=PAM:authentication acct="extreme" exe="/usr/sbin/sshd"
hostname=192.168.10.12 addr=192.168.10.12 terminal=ssh res=success'
type=USER_ACCT msg=audit(1591975763.219:41): pid=7894 uid=0 auid=4294967295
ses=4294967295 msg='op=PAM:accounting acct="extreme" exe="/usr/sbin/sshd"
hostname=192.168.10.12 addr=192.168.10.12 terminal=ssh res=success'
type=CRED_ACQ msg=audit(1591975763.223:42): pid=7894 uid=0 auid=4294967295 ses=4294967295
msg='op=PAM:setcred acct="extreme" exe="/usr/sbin/sshd" hostname=192.168.10.12
addr=192.168.10.12 terminal=ssh res=success'
type=LOGIN msg=audit(1591975763.223:43): pid=7894 uid=0 old-auid=4294967295 auid=1000
tty=(none) old-ses=4294967295 ses=127 res=1
type=USER_START msg=audit(1591975764.487:44): pid=7894 uid=0 auid=1000 ses=127
msg='op=PAM:session_open acct="extreme" exe="/usr/sbin/sshd" hostname=192.168.10.12
addr=192.168.10.12 terminal=ssh res=success'
type=CRED_ACQ msg=audit(1591975764.491:45): pid=8113 uid=0 auid=1000 ses=127
msg='op=PAM:setcred acct="extreme" exe="/usr/sbin/sshd" hostname=192.168.10.12
addr=192.168.10.12 terminal=ssh res=success'
type=USER_LOGIN msg=audit(1591975764.547:46): pid=7894 uid=0 auid=1000 ses=127
msg='op=login id=1000 exe="/usr/sbin/sshd" hostname=192.168.10.12 addr=192.168.10.12
terminal=/dev/pts/1 res=success'
```

Further, when the 'extreme' user authenticates to the 'XCO' command line with `efa login`, the following audit trail message is generated (in case of success):

```
type=USER_AUTH msg=audit(1591975780.823:47): pid=21139 uid=0 auid=4294967295
ses=4294967295 msg='op=PAM:authentication acct="extreme" exe="/apps/bin/hostauth"
hostname=? addr=? terminal=? res=success'
```

And here is the output in case of failures:

```
type=USER_AUTH msg=audit(1591976323.760:58): pid=21139 uid=0 auid=4294967295
ses=4294967295 msg='op=PAM:authentication acct="extreme" exe="/apps/bin/hostauth"
hostname=? addr=? terminal=? res=failed'
```

Secure NTP and XCO

For XCO, authenticated NTP can be used in two possible configurations:

- XCO acts as a client to existing NTP infrastructure.
- XCO runs its own NTP server. Steps for achieving this solution are provided below.

Also, there are many different visions for secure NTP that can range from authentication (at the low end) to leveraging the newest NTS (Network Time Secure) protocol that leverages public key cryptography via TLS (at the high end).

XCO is based on Ubuntu 18.04, and `ntpsec` from the upstream package maintainers does not allow for a comprehensive and well-supported usage of NTS. However, if this is an absolute requirement, you can achieve NTS support on XCO by manually compiling and deploying a recent version of `ntpsec`, **but this is not a supported solution**. This comes at the cost of not using the sanctioned Ubuntu packaging system for package updates, and this tradeoff may not be worth it within certain operational environments.

Enforcing authentication where XCO acts as a client to existing NTP infrastructure

If XCO is to act only as a client for authenticated NTP, then upstream NTP servers where XCO is pointed also need to support authenticated NTP. You can use public NTP servers for this purpose, such as those of the US National Institute of Standards and Technology (NIST). For more information, see the [NIST website](#). The following is an excerpt from the site:

"The time messages will be authenticated using symmetric-key encryption in a manner that is fully compatible with the published NTP documentation. (Autokey and asymmetric key modes will not be used.) Each registered user will be assigned a unique encryption key, which will be linked to the IP address(es) of the user's system(s).

A registered user will be able to communicate with the authenticated server using this assigned encryption key or using a default key of 0, which is equivalent to disabling the encryption algorithm. Users who are not registered will not be able to connect to this server, but can use any of the other NIST servers, which will not be modified."

Enforcing authentication where XCO provides its own NTP server

1. Populate the `/etc/ntp.keys` file with a SHA1 symmetric key. This can easily be done with the following command (which uses the `/etc/shadow` file as input to produce the SHA1 hash, and this hash must be shared with all clients that need to authenticate):

```
echo "3 SHA1 `shasum /etc/shadow |cut -d ' ' -f 1`" > ntp.keys
```

2. Add the line `'keys /etc/ntp.keys'` to the `/etc/ntp.conf` file on XCO.
3. Share the SHA1 symmetric key from Step 1 with any clients that are authorized to acquire NTP data from XCO. This could include the population of managed SLX-OS devices for example.
4. Restart the NTP daemon on XCO, and verify that no start-up error for the `ntpd` process is logged to `/var/log/syslog`. For example, if the format of the `/etc/ntp.keys` file is invalid, then the following error is displayed.

```
Jun 12 20:38:36
tpvm ntpd[16498]: authreadkeys: rejecting file '/etc/ntp.keys' after 1 error(s)
```



Note

From the perspective of NTP clients, if coming from a Linux system the `/etc/ntp.keys` file from the server configuration steps above should be available in the file system. For example, if this file is also placed at `/etc/ntp.keys`, then the step for acquiring NTP information from XCO is as follows:

```
# sntp -k /etc/ntp.keys <efa hostname>
```

Secure DNS and XCO

To achieve secure DNS communications using `dnscrypt-proxy`, perform the following steps.

Achieving secure DNS communications can be a critical aspect of a strong operational security posture. Use the open source 'dnscrypt-proxy' package to bring encrypted DNS communications to XCO.

1. Install dnscrypt-proxy by running:

```
apt-get install
    dnscrypt-proxy
```

2. Review the various resolvers that support encrypted DNS in the file `/usr/share/dnscrypt-proxy/dnscrypt-resolvers.csv` and select one. The name of this resolver is in the first column in this file. For example, the 'adguard-dns-ns1' resolver is displayed here:

```
adguard-dns-ns1,"Adguard DNS 1","Remove ads and protect your computer from
    malware","Anycast","",https://adguard.com/en/adguard-dns/
overview.html,1,no,yes,no,176.103.130.130:5443,2,dnscrypt.default.ns1.adguard.com,D12B:
47F2:52DC:F2C2:BBF8:9910:86EA:F79C:E449:5D8B:16C8:A0C4:322E:52CA:3F39:0873,pk.default.n
s1.adguard.com
```

3. Edit the `/etc/dnscrypt/dnscrypt-proxy.conf` file, and set

```
ResolverName adguard-dns-ns1

LocalAddress 127.0.0.1:53
```

4. Edit the `/etc/systemd/system/sockets.target.wants/dnscrypt-proxy.socket` file and make sure the `[Socket]` section looks like this:

```
[Socket]
    ListenStream=127.0.0.1:53
    ListenDatagram=127.0.0.1:53
```

5. Reload systemd, disable and re-enable dnscrypt-proxy.socket, and reboot:

```
# systemctl daemon-reload
# systemctl disable dnscrypt-proxy.service
# systemctl enable dnscrypt-proxy.service
# reboot
```

At this point, dnscrypt-proxy should be functioning normally. One way to verify this is to look for DNS requests on the XCO management interface ('eth0' for TPVM installations of XCO). There should be no traffic on UDP port 53 because encrypted DNS traffic is sent over UDP port 443.

6. To verify, run a sniffer on eth0, cause the system to issue a DNS lookup, and ensure that there are no UDP packets on port 53. The output should look similar to this:

```
# tcpdump -i eth0 -l -nn port 53 or port 443
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

15:26:54.136556 IP 192.168.10.109.40632 > 176.103.130.130.443: UDP, length 512
15:26:54.151425 IP 176.103.130.130.443 > 192.168.10.109.40632: UDP, length 304
15:26:54.152783 IP 192.168.10.109.40632 > 176.103.130.130.443: UDP, length 512
15:26:54.166523 IP 176.103.130.130.443 > 192.168.10.109.40632: UDP, length 304
```

Detect Rootkits with rkhunter

Rootkit Hunter (rkhunter) is a standard tool for the detection of rootkits on Linux. The Ubuntu Linux distribution maintains a package for rkhunter. The complete set of checks that rkhunter performs provides a good security baseline for finding some of the most malicious elements

of the offensive security landscape. It is recommended you regularly run the `rkhunter --check` command and review the contents of the `/var/log/rkhunter.log` file.

1. Install rkhunter by running:

```
# apt-get install rkhunter
```

2. Run a scan for rootkits:

```
# rkhunter --check
T0rn Rootkit           [ Not found ]
trNkit Rootkit         [ Not found ]
Trojanit Kit           [ Not found ]
Tuxendo Rootkit        [ Not found ]
URK Rootkit            [ Not found ]
Vampire Rootkit        [ Not found ]
VcKit Rootkit          [ Not found ]
Volc Rootkit           [ Not found ]
Xzibit Rootkit         [ Not found ]
zaRwT.KiT Rootkit      [ Not found ]
ZK Rootkit             [ Not found ]
```

3. For additional details of what is being checked on the system, refer to the `/var/log/rkhunter.log` file. For example, in the following example, the scan looked for evidence of the T0rn rootkit and specifically, the existence of the following files were checked (output abbreviated):

```
[21:28:18] Checking for T0rn Rootkit...
[21:28:18] Checking for file '/dev/.lib/lib/lib/t0rns' [ Not found ]
[21:28:18] Checking for file '/dev/.lib/lib/lib/du' [ Not found ]
[21:28:18] Checking for file '/dev/.lib/lib/lib/ls' [ Not found ]
[21:28:18] Checking for file '/dev/.lib/lib/lib/t0rnsb' [ Not found ]
[21:28:18] Checking for file '/dev/.lib/lib/lib/ps' [ Not found ]
[21:28:18] Checking for file '/dev/.lib/lib/lib/t0rnp' [ Not found ]
[21:28:18] Checking for file '/dev/.lib/lib/lib/find' [ Not found ]
[21:28:18] Checking for file '/dev/.lib/lib/lib/ifconfig' [ Not found ]
[21:28:18] Checking for file '/dev/.lib/lib/lib/pg' [ Not found ]
[21:28:18] Checking for file '/dev/.lib/lib/lib/ssh.tgz' [ Not found ]
[21:28:18] Checking for file '/dev/.lib/lib/lib/top' [ Not found ]
[21:28:19] Checking for file '/dev/.lib/lib/lib/sz' [ Not found ]
[21:28:19] Checking for file '/dev/.lib/lib/lib/login' [ Not found ]
[21:28:19] Checking for file '/dev/.lib/lib/lib/in.fingerd' [ Not found ]
[21:28:19] Checking for file '/dev/.lib/lib/lib/li0n.sh' [ Not found ]
```

Configure Remote Syslog (RELP) Securely

You can configure remote Syslog (RELP) securely.

Follow this procedure to configure remote Syslog (RELP) securely.

1. Install certtool on the Ubuntu system.

```
sudo apt-get install gnutls-bin
```

2. Extract `relp-certs.tar.gz` and edit `ca.cfg` and `server.cfg` per guidelines (`client.cfg` is not needed if you are configuring the client from XCO).



Note

Download the tar file `relp-certs.tar.gz`.

For guidelines on updating `ca.cfg` and the `server.cfg` files, see [Configuration Files Modification Guidelines](#) on page 100.

3. Run the `./1-generate-ca.sh` command to generate CA files (ca.pem and ca-key.pem).
4. Run the `generate-server.sh` command to generate server certificates (server-cert.pem and server-key.pem).
5. Copy extreme-ca-chain.pem from EFA's /apps/efadata/certs/ca location to syslog server's /opt/notification/ location (create folder if absent).

**Note**

If a third-party or CMPv2 client certificate is installed on XCO, copy the third-party or CMPv2 CA chain that signed the certificate instead of the extreme-ca-chain.pem file.

6. Copy server-cert.pem and server-key.pem to /opt/notification/.
7. Edit /etc/rsyslog.conf with root/sudo permission. Add the following lines above GLOBAL DIRECTIVES in the file:

```
module(load="imrelp")

input(type="imrelp" port="20514" tls="on" tls.caCert="/opt/notification/extreme-ca-chain.pem" tls.myCert="/opt/notification/server-cert.pem" tls.myPrivKey="/opt/notification/server-key.pem" tls.authMode="name" tls.permittedpeer=["tpvm128", "10.20.63.128", "10.20.48.245", "10.20.246.7", "10.20.241.7", "10.20.63.70", "tpvm-72", "administrator-00", "efa.extremenetworks.com", "10.20.54.90"])
```

**Note**

- Ensure to include the XCO VIP address or the IP address/server address (where XCO is installed) in the list of IP addresses.
- If you used a third-party or CMPv2 CA chain in Step 5, set tls.caCert to that file instead of extreme-cachain.pem.

8. Restart rsyslog service.

```
sudo systemctl restart rsyslog
```

9. Complete the following RELP configuration in XCO:
 - a. Copy ca.pem from syslog server to XCO's /home/extreme.
 - b. Add subscriber.

```
efa notification subscribers add-syslog-relp --address=<syslog-IP>:20514 --cacert=ca.pem --conn-timeout=30 (for example, 10.21.88.25:20514)
```

**Note**

Address is the Notification Server IP address.

- c. Run some of the XCO commands to verify syslog receipt (if the messages are forwarded).

On XCO:

```
(efa:xmcdev)xmcdev@xmcdev-virtual-machine1:/opt/checkouts/efa/efa$ efa tenant create --name=tnt3

Tenant created successfully.

--- Time Elapsed: 209.27997ms ---

<5>Apr  6 01:49:35 10.133.131.162(10.133.131.162)
```

```
{ "application": "ts", "device_ip": "", "message": "Tenant create request  
received :request={\"name\\\": \"tnt3\\\"}\", \"scope\": \"user\", \"status\": \"started\", \"task\": \"EFA-001001\", \"timestamp\": \"2022-04-06T01:49:04+05:30\", \"type\": \"Task\", \"username\": \"xmcdev\"}  
  
<5>Apr 6 01:49:35 10.133.131.162(10.133.131.162)  
{ \"application\": \"ts\", \"device_ip\": \"\", \"message\": \"Tenant create request  
success :request={\"name\\\": \"tnt3\\\"}\", \"scope\": \"user\", \"status\": \"succeeded\", \"task\": \"EFA-001002\", \"timestamp\": \"2022-04-06T01:49:04+05:30\", \"type\": \"Task\", \"username\": \"xmcdev\"}
```

Configuration Files Modification Guidelines

Learn modification guidelines for `ca.cfg` and `server.cfg` files.

ca.cfg

Uncomment or update the following values:

- `organization = "Extreme Networks"`
- `unit = "Extreme"`
- `country = US`
- `cn = "Extreme CA"`
- `expiration_days = 3650`
- `challenge_password = (leave empty)`
- `ca`
- `signing_key`
- `cert_signing_key`

server.cfg

Uncomment or update the following values:

- `organization = "Extreme Networks"`
- `unit = "Extreme"`
- `cn = "ana-sowj.extremenetworks.com"`
- `expiration_days = 3650`
- `ip_address = "10.21.88.25"`
- `challenge_password = (leave empty)`
- `tls_www_client`
- `tls_www_server`