Extreme SLX-OS
Security Hardening Guide, 20.4.1

Supporting ExtremeRouting and ExtremeSwitching
SLX 9740, SLX 9640, SLX 9540, SLX 9250,
SLX 9150, Extreme 8720, and Extreme 8520

9037418-00 Rev AA
April 2022

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: www.extremenetworks.com/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/

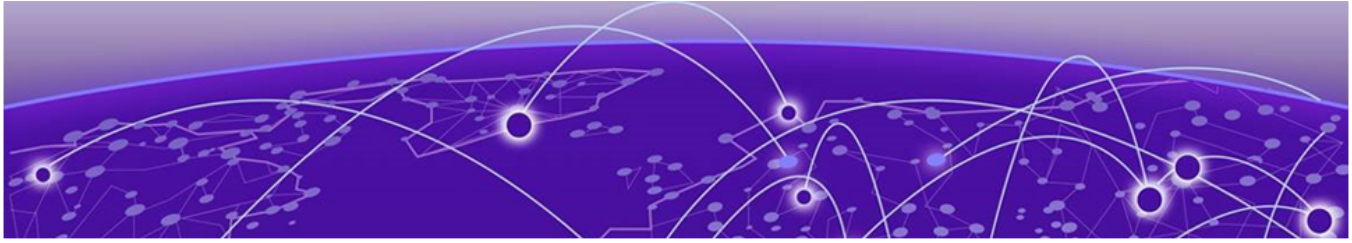


Table of Contents

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

Table 1: Notes and warnings

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

Table 2: Text

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

Table 3: Command syntax

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

Documentation and Training

Find Extreme Networks product information at the following locations:

- Current Product Documentation
- Release Notes
- Hardware and software compatibility for Extreme Networks products
- Extreme Optics Compatibility
- Other resources such as white papers, data sheets, and case studies

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For details, visit www.extremenetworks.com/education/.

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 2826. 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 Information Development 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:

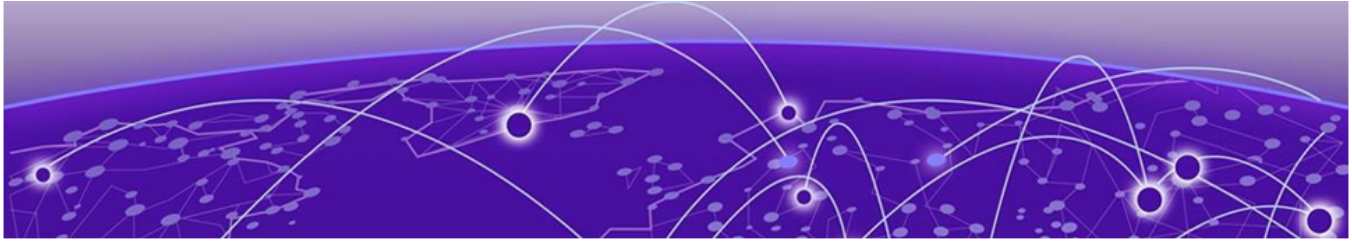
- Content errors, or confusing or conflicting information.

- Improvements that would help you find relevant information.
- Broken links or usability issues.

To send feedback, do either of the following:

- Access the feedback form at https://www.extremenetworks.com/documentation-feedback/.
- Email us at documentation@extremenetworks.com.

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.



About this document

What's New in this Document

The following table includes descriptions of new information added to this document for the SLX-OS 20.4.1 software release.

Table 4: Summary of changes

Feature	Description	Described In
Security Enhanced Linux	SLXOS now supports SE Linux	Security Enhanced Linux (SE Linux) on page 25

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

Supported Hardware

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

SLX-OS 20.4.1 supports the following hardware platforms.

- Devices based on the Broadcom XGS® chipset family:
  - Extreme 8720
  - Extreme 8520
  - ExtremeSwitching SLX 9250
  - ExtremeSwitching SLX 9150
- Devices based on the Broadcom DNX® chipset family:
  - ExtremeRouting SLX 9740
  - ExtremeRouting SLX 9640

- ExtremeSwitching SLX 9540

Note

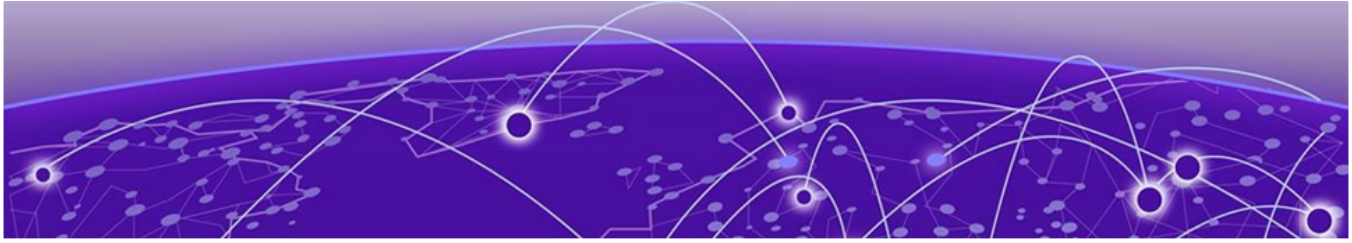
All configurations and software features that are applicable to SLX 9150 and SLX 9250 devices are also applicable for the Extreme 8520 and Extreme 8720 devices respectively.

The "Measured Boot with Remote Attestation" feature is only applicable to the Extreme 8520 and Extreme 8720 devices. It is not supported on the SLX 9150 and SLX 9250 devices.

Note

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

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



Security Hardening Guide

Security Hardening Guidance Overview

This section describes suggested configuration actions to harden the Extreme Networks OS switch.

Device hardening steps are performed by a user with administrative privileges. Specific hardening actions may or may not be appropriate for a given environment and must be considered in the context of the overall security policy and existing physical and procedural controls.

The NetworkOS device management functions are isolated through authentication. Once administrators login with specific credentials, their access is limited to commands for which they have privileges with role-based permissions. Additionally, network management communication paths are protected against modification and disclosure using SSHv2. The audit channel to an external Syslog server is protected using TLS encapsulation.

Configure password policies

This section details on how to configure the password policies.

The minimum password strength and configurable attributes are recommended that includes minimum length, character sets, with the number of retries when logging in. This record details on how long an account can be locked out when the maximum number of login failures is observed.

An example password policy configuration:

```
device(config)# password-attributes min-length 8
device(config)# password-attributes max-retry 4
device(config)# password-attributes max-lockout-duration 5000
device(config)# password-attributes character-restriction upper 1
device(config)# password-attributes character-restriction lower 2
device(config)# password-attributes character-restriction numeric 1
device(config)# password-attributes character-restriction special-char 1
```

The default password encryption policy is Encryption Level 10, which utilizes salted SHA512 for password storage.

Refer to the Extreme SLX-OS Security Configuration Guide, 20.3.3, topic *Password Policies* for further details.

Administrator lockout

This topic provides information about the administrator locout details.

By default, the administrator is not locked out of the device even after `max-retry` failures. To lock the administrator out, execute the below command:

```
device(config)# password-attributes admin-lockout
```

When the administrator is locked-out, the device allows access for the administrator after the value set for `max-lockout-duration` has elapsed.

Note
The administrator logins over the serial port/console, which is never locked out and can login over the network again only if the admin-lockout password attribute is disabled.

To allow the administrator to login over the network and disable administrator lockout execute the below commands:

```
device# configure terminal

device(config)# no password-attributes admin-lockout
```

Reference the Extreme SLX-OS Security Configuration Guide, 20.2.1 – Password Policies for further details

SSH Configuration

SSH ciphers

The following ciphers are recommended for the SSH client and SSH server:

- `aes256-ctr`
- `aes256-cbc`

Reference the Extreme SLX-OS Security Configuration Guide, 20.2.1 – Configure SSH Ciphers for specific guidance configuring SSH ciphers.

SSH MAC algorithms

The following MAC algorithms are recommended for the SSH client and SSH server:

- `hmac-sha2-256`
- `hmac-sha2-512`

Reference the Extreme SLX-OS Security Configuration Guide, 20.2.1 – Configure SSH MAC for specific guidance configuring SSH MAC algorithms.

SSH Key-exchange

The following MAC algorithms are recommended for the SSH client and SSH server:

- `ecdh-sha2-nistp256`
- `diffie-hellman-group14-sha1`

Reference the Extreme SLX-OS Security Configuration Guide, 20.2.1 – Configure SSH Key-exchange for specific guidance configuring SSH Key-exchange algorithms.

SSH server timeout and login policies

Enter the `ssh` server max-idle-timeout command to set the timeout value for SSH connections to the server. This setting affects `ssh` connections to the server including the `netconf` sessions.

```
device(config)# ssh server max-idle-timeout 20
```

Enter the **`sshserver max-auth-tries`** command to set the number of login attempts

```
device(config)# ssh server max-auth-tries 2
```

Enter the **`sshserver max-login-timeout`** command to set the login timeout. Set the value to an appropriate timeout period in the administrator's environment.

```
device(config)# ssh server max-login-timeout 30
```

ConfiguringSSH session re-key interval by volume and time

The SSH servers can trigger re-keying once a certain time interval is reached or data traffic reaches a specified volume. During re-keying, a set of key exchange messages are transferred between the SSH client and the server, changing the key used for the session security.

Re-keying by volume

The re-key-volumeoption cannot exceed a value equal to `1024` MB. The default value is `1024` MB. The range of the rekey volume configured using the ssh-server command is `512` to `1024` MB.

```
device(config)# ssh server rekey-volume ? Possible completions:
```

```
<DECIMAL> <512-4095> Megabytes"
```

Re-keying by time

The SSH rekey can also be configured based on time. The default value is `3600` seconds. The following command is used to specify the time.

```
device(config)# ssh server rekey-interval ?Possible completions:

<DECIMAL> <900-3600> Seconds
```

Configure SSH authentication method

The SSH provides public key and password authentication methods, including support for `X.509 v3` certificates.

To use SSH public-key authentication, enter the **`certutil import sshkey directory`** *pubkey-directory*file*filename*protocol SCP host *remote-ip* user *user-account password password* command to import the public key.

```
device# certutil import sshkey user admin host 10.70.4.106 directory /
users/home40/bmeenaks/.ssh file id_rsa.pub login fvt

Password: ***********
2012/11/14-10:28:58, [SEC-3050], 75,, INFO, VDX, Event: sshutil, Status:
success, Info: Imported SSH public key from 10.70.4.106 for user
'admin'.
```

To support password less SSH authentication, externally generated key pairs using `RSA-2048`.

Reference the Extreme SLX-OS Security Configuration Guide, 20.2.1 – Secure Shell for further guidance configuring SSH authentication method.

Disable telnet server

Enterthe telnet server shutdown command in global configuration mode to disable the Telnet server.

```
device(config)# telnet server shutdown
```

Disable TLS 1.1 and older

This topic details the procedure to disable TLS 1.1 and older versions.

1. SSH to the system and acquire a root shell:
   ```
   SLX# start shell
   Entering Linux shell for the user: admin
   [admin@SLX]# su -
   Password:
   [root@SLX]#
   ```
2. Edit the `Apache webserver` config located at '`/fabos/webtools/bin/web.conf.0`' and replace the line that contains the '`SSLProtocol`' variable with the following:
   ```
   SSLProtocol -all +TLSv1.2
   ```

3. Grep the process table to look for `httpd` processes and kill the lowest numbered one (first in the list). For example:

```
# ps axuww |grep httpd
nobody    5046  0.0  0.0  88956  4220 ?        S    20:32   0:00
        /usr/sbin/httpd.0 -DSSL -f /fabos/webtools/bin/httpd.conf.0
root     24164  0.0  0.0  88688  6360 ?        Ss   01:59   0:14
        /usr/sbin/httpd.0 -DSSL -f /fabos/webtools/bin/httpd.conf.0
nobody   29385  0.0  0.0  88956  4220 ?        S    19:22   0:00
        /usr/sbin/httpd.0 -DSSL -f /fabos/webtools/bin/httpd.conf.0
# kill 5046
```

4. Restart `Apache` by manually executing the following command:

```
# /usr/sbin/httpd.0 -DSSL -f /fabos/webtools/bin/httpd.conf.0
```

5. At this point, `SLX-OS` will be running `Apache` with `TLS < 1.2` disabled.

Note

The **`httpd.conf.0`** file includes the **`web.conf.0`** file automatically and there is no persistent change across reboots. However, this will be fixed in future SLX-OS release.

Enable authentication services

This section details the procedure to enable the authentic services of HTTPS, TLS, and SYSLOG.

Enable HTTPS

Reference the Extreme SLX-OS Security Configuration Guide, 20.2.1 – HTTPS Certificates for specific guidance on installing certificates and enabling HTTPS.

Enable TLS for remote authentication services

RADIUS over TLS and LDAP over TLS are supported.

Reference the Extreme SLX-OS Security Configuration Guide, 20.2.1 – RADIUS Server Authenticationfor specific guidance on configuring RADIUS over TLS.

Reference the Extreme SLX-OS Security Configuration Guide, 20.2.1 – Lightweight Directory Access Protocol for specific guidance on configuring LDAP over TLS.

Enable TLS for SYSLOG

To enable secure logging using the `syslog` server, complete the following steps.

1. Enter the **crypto import syslogca** command in privileged EXEC mode to import the syslog CA certificate.

```
device# crypto import syslogca rbridge-id 1 protocol SCP host 10.2.2.101 directory
        /home/certs/ file chainCA02.cert.pem user admin password <password>
```

The CA certificate imported must be generated using `RSA-2048` with `SHA-256`.

- Enter the **logging syslog-server** *ip-address* command in global configuration mode to configure the syslog server.

```
device(config)# logging syslog-server 10.20.238.120
        secure port 1999
```

The device enforces certificate validation during import and TLS server certificate validation occurs during the TLS handshake according to the following rules:

- Certificate validation and the certificate path validation support a minimum path length of three certificates.
- The certificate path must terminate with a trusted CA certificate.
- The certificate path should be validated by verifying the presence of the `basic Constraints`extension and that the CA flagis setto TRUE for all CA certificates.
- The revocation status of the certificate should be validated.
- For `SYSLOG`, the device currently requires that an IP address must be used for Common Name (CN) and Subject Alternative Name (SAN).

The `extendedKeyUsage` field should be validated according to the following rules:

- Certificates used for trusted updates and executable code integrity verification should have the Code Signing purpose `(id- kp 3 with OID 1.3.6.1.5.5.7.3.3)` in the `extended Key Usage`field.
- Server certificates presented for TLS should have the Server Authentication purpose `(id-kp 1 with OID 1.3.6.1.5.5.7.3.1)` in the `extended Key Usage` field.
- Client certificates presented for TLS should have the Client Authentication purpose `(id-kp 2 with OID 1.3.6.1.5.5.7.3.2)` in the `extended Key Usage` field.
- OCSP certificates presented for OCSP responses should have the OCSP Signing purpose `(id-kp 9 withOID 1.3.6.1.5.5.7.3.9)` in the `extended Key Usage` field.
- A certificate should only be treated as a CA certificate if the `basic Constraints` extensionis present and the CA flag is set to TRUE.

Disable unused remote authentication services

This section details the method to disable unused remote authentic services.

Enterthe **`no tacacs-server`** command to remove any TACACS + server configuration.

```
device(config)# no tacacs-server <host>
```

Enterthe **`no radius-server`** command to remove any RADIUS server configuration.

```
device(config)# no radius-server <host>
```

Enterthe **`no ldap-server`** command to remove any LDAP server configuration.

```
device(config)# no ldap-server <host>
```

Configure IP ACLs to block services

This section details to configure IP ACLs.

Use IP ACLs to block Telnet, HTTP, and Extreme internal ports `7110, 7710, 8008, 9110,` and `9710` for `IPv4` and `IPv6`. If SSH access is required, enter **`seq permit`** commands to allow access on port 22.

If remote access is required, such as through SCP or LDAP, enter **seq permit** commands to allow UDP and TCP traffic on ports 1024 through 65535.

Configure IP ACLs usingthe **ip access-list** command and use the **ip access-group** command to apply the rules to the management interface.

```
device(config)# ip access-list extended ccextACL
device(config-ip-ext)# seq 1 deny tcp any any eq 23
device(config-ip-ext)#seq 2 deny tcp any any eq 80
device(config-ip-ext)#seq 5 deny tcp any any eq 7110
device(config-ip-ext)#seq 6 deny tcp any any eq 7710
device(config-ip-ext)#seq 7 deny tcp any any eq 8008
device(config-ip-ext)#seq 8 deny tcp any any eq 9110
device(config-ip-ext)#seq 9 deny tcp any any eq 9710
device(config-ip-ext)#seq 11 permit tcp any any range 1024 65535
device(config-ip-ext)#seq 12 permit udp any any range 1024 65535
device(config-ip-ext)#seq 13 permit tcp any any eq 22
device(config-ip-ext)#seq 14 permit tcp any any eq 830
device(config-ip-ext)#exit
device(config)# interface management 1/0
device(config-Management-1/0)# ip access-group ccextACL in

device(config)# ipv6 access-list extended ccextACL6
device(config-ip-ext)# seq 1 deny tcp any any eq 23
device(config-ip-ext)#seq 2 deny tcp any any eq 80
device(config-ip-ext)#seq 5 deny tcp any any eq 7110
device(config-ip-ext)#seq 6 deny tcp any any eq 7710
device(config-ip-ext)#seq 7 deny tcp any any eq 8008
device(config-ip-ext)#seq 8 deny tcp any any eq 9110
device(config-ip-ext)#seq 9 deny tcp any any eq 9710
device(config-ip-ext)#seq 11 permit tcp any any range 1024 65535
device(config-ip-ext)#seq 12 permit udp any any range 1024 65535
device(config-ip-ext)#seq 13 permit tcp any any eq 22
device(config-ip-ext)#seq 14 permit tcp any any eq 830
device(config-ip-ext)#exit
device(config)# interface management 1/0
device(config-Management-1/0)# ipv6 access-group ccextACL6 in
```

Reference the Extreme SLX-OS Security Configuration Guide, 20.2.1 – ACLs for specific guidance.

```
device(config)# no ldap-server <host>
```

Configure banners

This section details on how to configure banners.

The commands below are used to configure banner messages. The banner messages are used to provide information to the user when the device is accessed. There are three commands that can be used to setup banner messages as explained below:

- banner incoming –sets the incoming banner message. The message is seen on the console when a user accesses the device.
- banner motd –sets the message of the day banner. The message is displayed when the device receives a login request. Also used to display a message for other users of the switch.
- banner login –sets the switch banner and the message is displayed after the user is authenticated.

Banner length is from 1 – 2048 characters, which can be issued as a single line of text, or in multiline mode by pressing esc m.

Each of these commands can be invoked from the CLI in configure mode as shown below:

```
device # banner incoming <message>
device # no banner incoming
device # banner motd <message>
device # no banner motd
device # banner login <message>
device # no banner login
```

Support for RSA 4096 bit SSH hostkey

This feature is introduced in SLX for hardening the security of management plane.

Releases from 20.3.2 and above provide support for the strongest of RSA hostkeys for SSH, which is 4096 bits. Prior to this, the maximum size supported for SSH RSA hostkey was 2048 bits. Now you can configure SSH RSA hostkey 4096 bit using below option.

```
SLX(config)# ssh server key rsa ?
Possible completions:
[2048]
1024 1024 bits RSA key
2048 2048 bits RSA key [default]
4096 4096 bits RSA key
SLX(config)# ssh server key rsa 4096
```

The default **RSA** hostkey for **SSH** when the above hostkey is not configured, is **2048** bits.

The SLX provides the **SSH** server hostkey algorithms **RSA, ECDSA P256** and **DSA** to be configured. It's recommended to use only **ECDSA** or **RSA**(minimum **2048** bits) as the hostkeys **DSA** and **RSA 1024** are both insecure with **1024** bit length.

The **ECDSA** being the strongest algorithm, the OpenSSH server in SLX sends this as the hostkey if it is present. On a linux, the SSH client receives the following message to accept the hostkey sent by the SLX.

```
The authenticity of host 10.24.12.129 (10.24.12.129) can't be established.
ECDSA key fingerprint is SHA256:LlgBLdBedpJ1AU6GWa4OYjtye6JM4CfR8i8k2SwGOfw.
Are you sure you want to continue connecting yes/no ?
```

If you remove ECDSA hostkey configured from SSH server key CLI , then the OpenSSH server in SLX negotiates **RSA** hostkey based on the bit length, which you configured using **ssh** server key **RSA** CLI.

The default being 2048 bits. Hence, you need to explicitly configure **ssh** server key **RSA 4096** to use the **RSA 4096** bit hostkey and remove **ECDSA** if it does not consider, so that the server sends **RSA 4096** as the hostkey.

Connlimit as an option for ip access-lists

This feature is introduced in SLX lately for hardening the security of management plane.

Starting from 20.3.2 release, management ACL configurations provides connlimit as an option to restrict the number of connections from a given host, using a specific protocol or application port.

For example- the below configuration when applied to the management port restricts the number of tcp connections to the SLX mgmt. IP from the given client IP to N, where N can range between *1–65536*.

SLX(config)#ip access-list extended check

```
SLX(conf-ipacl-ext)# permit tcp host <client IP> host <SLX mgmtIP>
connlimit <N>
```

The protocol option **tcp** in the above example is specified as **udp**, which restricts **udp** connections or can be specified as even **ip** that restricts both **udp** and **tcp** connections to given number N.

Instead of specific client IP there is an option to provide **any** as an argument that filter connections on specified protocol from any IP address to the given value N for the connection limits.

```
SLX(conf-ipacl-ext)# permit ip any any connlimit N
```

```
SLX(conf-ipacl-ext)# permit tcp any any connlimit N
```

Note

Only incoming connections on management port can be restricted using the connlimit option.

When we specify application ports to the access-list with conn limit option, you can apply these restriction to specific application protocols alone and allow rest of the traffic. The below example restricts number of SSH connections, which uses port 22 from the specified client IP.

```
SLX(conf-ipacl-ext)# permit tcp host <client IP> host <SLX mgmtIP> eq 22
connlimit N.
```

Restricting connlimits to application protocols can be a highly useful service in mitigating DDOS attacks, by preventing access from malicious clients.

For example - the HTTP/HTTPs service in SLX has a restriction of allowing maximum of 30 parallel REST connections to SLX.

A DDOS attack scenario is observed in field, when a buggy client, which do not close its sockets but monitor SLX heart beat that sends periodic unauthorized REST requests to SLX at a rapid rate, such that it exhausted the 30 connections in quick time and denied the other operational REST client access to SLX for sometime, until the socket state transitions cleaned up the orphaned connections.

The connlimit can be applied in the below manner to prevent such a DDOS attack from a malicious client by specifying the connlimit like below, if it is known that the operational REST client would have maximum of say 10 parallel REST connections to SLX.

In the below example client IP 1 is the operational client and client IP 2 is the heartbeat monitor. It also prevents a malicious client IP 3 from exhausting the REST connections in similar way and causing denial of service.

```
SLX(conf-ipacl-ext)# permit tcp host <client IP 1> host <SLX mgmtIP> eq
https connlimit 10
```

```
SLX(conf-ipacl-ext)# permit tcp host <client IP 2> host <SLX mgmtIP> eq
https connlimit 10
```

To use the `connlimit` options with an access-list it may be required that the administrator first configures a `permit ip` rule with `any any` option to allow other traffic without disruption. This is because of the order of IP table rules that are added by default in SLX.

In order to understand the IP tables order and how to use `connlimit` correctly and the limitations, please refer to 20.3.2 security guide ACL section.

It is also suggested that administrators do not use `connlimit` with generic protocols like `tcp`, `udp` and `ip` with `any any` option, unless they are very familiar with management connections to the SLX on field to prevent possible disruption of traffic that is not intended to be restricted.

Version control for TLS

This feature is introduced in SLX lately for hardening the security of management plane.

Starting from 20.3.2 release, it's possible for the administrator to configure the minimum TLS protocol version to be used by SLX manageability applications that use TLS either as a client or server. The SLX provides separate TLS version control options for TLS clients and servers. The applications that act as TLS clients in SLX are `SYSLOG`, `RADIUS` and `LDAP`. The TLS servers of SLX management plane are HTTPs and secure GNMI. The control knobs are like below.

```
ssl-profile-server)# tls min-version ?
Possible completions:
<1.1|1.2> specify TLS version
SLX(mgmt-sec-ssl-profile-server)# tls min-version 1.2
SLX(mgmt-sec-ssl-profile-server)# exit
SLX(mgmt-security)# ssl-profile ?

Possible completions:
client management security ssl profile client for tls
configuration
server management security ssl profile server for tls
configuration

SLX(mgmt-security)# ssl-profile client
SLX(mgmt-sec-ssl-profile-client)# tls ?
Possible completions:
min-version min version to be supported by client
SLX(mgmt-sec-ssl-profile-client)# tls min-version ?

Possible completions:
<1.1|1.2> specify TLS version
SLX(mgmt-sec-ssl-profile-client)# tls min-version 1.2
SLX(mgmt-sec-ssl-profile-client)# end
SLX#
```

The basic intention of having the version control configuration for TLS is to encourage administrators to set the minimum version as TLS v1.2, which is the strongest secure TLS version that is supported in SLX.

Setting the minimum version to TLS v1.2 for the client profile forces TLS clients to send only TLS v1.2 version in its client hello packet as TLSv1.2 is the max supported TLS version in SLX. In case the server negotiates a lesser secure version, the SLX breaks the handshake upon receiving the server hello.

The below example is audit log that appears when the handshake is broken due to whichr the show logging audit command output indicates the insecure version that was negotiated by the server.

```
63 AUDIT, 2021/02/17-16:25:24 (GMT), [SEC-3111], INFO, SECURITY,
NONE/root/NONE/None/CLI,, SLX, Event: TLS SESSION, TLS handshake,
Info: server version 1.1 is lesser than client min-version 1.2 TLS
handshake failed.
```

Setting the minimum version to `TLS v1.2` for the server profile forces TLS servers in SLX to break the handshake upon receiving a client `hello` with less secure TLS version.

The below example is audit log that appears when the handshake is broken, due to which the show logging audit command output indicates the insecure version that was sent by the client.

```
63 AUDIT, 2021/02/17-16:25:24 (GMT), [SEC-3111], INFO, SECURITY,
NONE/root/NONE/None/CLI,, SLX, Event: TLS SESSION, TLS handshake,
Info: client version 1.1 is lesser than server min-version 1.2 TLS
handshake failed.
```

Note

Configuring the version control option or configuring TLS v1.1 would result in the same behaviour because by default the minimum supported TLS version in SLX is TLS v1.1.

Securing GNMI

This feature is introduced in SLX lately for hardening the security of management plane.

Starting from 20.3.2 administrators were able to protect GNMI with TLS. To use this service admin performs the following configuration.

`SLX(config-gNMI-server)# secure-port <port number>`

where the port number can vary from `1024 to 49151`. When this configuration is done GNMI runs over TLS on the above mentioned port and client get connected to this port to make a TLS connection.

The administrator need to use a GNMI client that has TLS support and configure it for the same.

Note

Removing the above configuration makes `GNMI` to switch to non -secure mode, which is default and will listen on the non-secure default `port 9339`.

On SLX which is the GNMI server, the GNMI server certificate and the private key signing it can be imported to the switch via `pkcs12` format just like HTTPs certificate and key are imported.

The following command option is provided for the same, where the certificate and the key is encrypted into `pkcs12` format file on a trusted external server and imported from that server.

```
SLX# crypto ca import-pkcs type pkcs12 cert-type gNMI-server directory
<dir-name> file <file-name> host <host-name/ip> protocol <SCP|FTP>
user <server-username> password <server-password> pkcs-passphrase
<pkcs export password>
```

Mutual authentication support for TLS

This feature is introduced in SLX lately for hardening the security of management plane.

From the release 20.3.3 the TLS clients and servers in SLX management plane are enabled with the ability to present, receive and validate client certificates to either authenticate itself or authenticate the remote client presenting its certificate to SLX.

Since the TLS clients on SLX are **syslog**, **RADIUS** and **LDAP** the two command options that are provided to import the `pkcs12` format of the client certificates and the private key signing it.

```
SLX# crypto ca import-pkcs type pkcs12 cert-type <ldap-client/radiusclient/
syslog-client> directory <dir-name> file <file-name> host
<host-name/ip> protocol <SCP|FTP> user <server username> password
<server-password> pkcs-passphrase <pkcs export password>
```

When the client certificate is imported to SLX via above command for each of the services, connecting to their servers via secure port that sends client certificates to the server only if the server requests for client certificate. Enabling the server to send client certificate request is external server configuration and not in the scope of the current document.

Since the TLS servers in SLX are HTTPs and secure GNMI, these must authenticate the external clients when the latter presents its client certificates. For this the TLS servers of SLX must send client certificate request in the TLS handshake.

To validate the incoming client certificate against a trusted authority during the TLS handshake, a CA certificate for the client certificate must be imported to SLX. The CA can be imported via the below command for HTTPs and secure GNMI.

To import CA of HTTPs client cert.

SLX#crypto import httpsclientca directory <dir-name> file <file-name> host <host-name/ip> protocol <SCP|FTP> user <server-username> password <server-password>

To import CA of GNMI client cert.

SLX#crypto import gnmiclientca directory <dir-name> file <file-name> host <host-name/ip> protocol <SCP|FTP> user <server-username> password<server-password>

Importing the CA via the above commands acts as a control knob for turning on mutual authentication and enables these services to request client certificate from clients during TLS handshake. When the client certificate is requested clients need to mandatorily present their client certificates issued by the imported CA.

Note

Removing the imported CA's via no form of the above commands disables mutual authentication for the respective services.

Enabling mutual authentication enhances security such that it prevents a man in the middle attack from imposing clients, which fail to identify themselves to SLX or to establish SLX as a trusted client to TLS servers seeking client authentication.

Certificate expiry alert levels and period configuration

This feature is introduced in SLX lately for hardening the security of management plane.

Starting from release 20.3.3 it's possible for the administrator to configure the number of days ahead of expiry of a TLS certificate present in SLX, as an alert to be issued from SLX and map this configured period against one of the 4 levels that indicate the severity of this alert. The 4 levels are critical, major, minor and info. Below is an example of how the configuration will be.

```
SLX(config)# crypto cert expiry-level info period 50
SLX(config)# crypto cert expiry-level minor period 30
SLX(config)# crypto cert expiry-level major period 10
SLX(config)# crypto cert expiry-level critical period 5
```

The configuration allows period to be specified in the range between `1 to 90` in number of days. It means when this configuration is done, periodically once in 24 hours the expiry date of all TLS certificates present in SLX are checked and when the number of days remaining for expiry of a certain certificate matches the period configured, an alert is issued with the severity indicating the level specified in the expiry-level field.

The alert is issued in the form of a RASLOG. On `SLX SNMP` trap severity levels can be set. Upon setting the SNMP trap severity level to warning the generated raslogs will also issue an SNMP trap.

The `RASLOGs` and `SNMP` traps carries detail about the expired certificate like the serial number of the certificate and its subject etc., and display that this certificate expires within these many days. Below is an example when info level is configured.

RASLOG

```
2022/05/13-00:00:02, [SEC-3136], 87,, WARNING, SLX, Event: cert expiry
,Alert-level:INFO, Certificate Details=[subject=
/C=IN/ST=KA/L=BAN/O=HCL/OU=Engg/CN=10.24.12.129/emailAddress=gselvara
j@hcl.com issuer= /C=IN/ST=KA/L=BAN/O=HCL/OU=Engg/CN=root
serial=4098] will expire in 44 days.
```

SNMP Trap

```
05:35:32.203670 IP 10.24.12.129.50000 > ldap.testsqa.com.SNMPtrap:
C="cm2" V2Trap(452) system.sysUpTime.0=81400
S:1.1.4.1.0=E:1588.2.1.1.1.0.4 S:18.1.3.0=10.24.12.129
E:1588.2.1.1.1.8.5.1.1.87=87 E:1588.2.1.1.1.8.5.1.2.87="2022/05/13-
00:00:02" E:1588.2.1.1.1.8.5.1.3.87=3 E:1588.2.1.1.1.8.5.1.4.87=1
E:1588.2.1.1.1.8.5.1.5.87="SEC-3136 Event: cert expiry , Alertlevel:
INFO, Certificate Details=[subject=
/C=IN/ST=KA/L=BAN/O=HCL/OU=Engg/CN=10.24.12.129/emailAddress=gselvara
j@hcl.com issuer= /C=IN/ST=KA/L=BAN/O=HCL/OU=Engg/CN=root
serial=4098] will expire in 44 days."
```

In case administrator has configured multiple levels or all 4 levels, then many alerts are issued indicating the particular severity level when the period remaining for expiry matches the configured period against each level.

In case a certificate is already expired a raslog with **Error as severity level** is sent continuously sent every 24 hours until the specific certificate is changed.

This RASLOG is sent irrespective of the expiry level configuration. Having the above configurations provides reminders to administrators to change the certificate and prevents a service from non-functional due to TLS handshake failure resulting from certificate expiry.

User account expiry period configuration upon inactivity

This feature is introduced in SLX lately for hardening the security of management plane.

Starting from release 20.3.3 administrators are able to configure the account expiry period for inactive users other than root and the default admin. It means if a user other than the root or default admin has not logged in for a period in number of days specified under the configuration as expiry period, then this user get automatically locked.

An administrator needs to explicitly unlocks this user to become active again. Unlocking the user resets the inactivity polling of the user account and polling starts again until the specified period. Resetting of the polling timer also happens when this user successfully logins before the inactivity period expires.

The configuration can be done for each user account when the user account is created or modified in SLX. In the below example the user Mike will get locked out if Mike hasn't logged in for 40 days and Mike will get a warning raslog alert if Mike hasn't logged in for 20 days.

```
SLX(config)#username Mike acct-inactivity-expiry-period 40 acctinactivity-
warning-period 20 password xyz@12345 role admin
```

The range for inactivity expiry period for an account can be configured between 1 to 180 days and the range for inactivity warning period is between 1 to 120 days.

Note
These configurations cannot be done to the root and default admin users, but can be done for the default user account or any other account.

The warning RASLOG is generated only once when the user do not login for the specified inactivity warning period.

Once the inactive user expires after the specified inactivity expiry period an error RASLOG indicating expiry will be sent every 24 hours. If the user configures SNMP trap severity level to warning these RASLOGs will also send an SNMP trap.

RASLOG Example

```
2021/03/04-09:50:00, [SEC-3138], 3445,, WARNING, SLX, Event: user
inactivity warning USER test will expire in 25 days.
2021/03/15-09:51:49, [SEC-3139], 3448,, ERROR, SLX, Event: user
expired USER test expired 12 days ago.
```

SNMP Trap Example

```
03:27:30.135220 IP 10.24.15.197.50000 > ldap.testsqa.com.SNMPtrap:
C="cm1" Trap(276) E:1588.2.1.1.1 10.24.15.197 enterpriseSpecific s=4
365800 S:18.1.3.0=10.24.15.197 E:1588.2.1.1.1.8.5.1.1.1918=1918
```

```
E:1588.2.1.1.1.8.5.1.2.1918="2020/12/26-02:53:09"
E:1588.2.1.1.1.8.5.1.3.1918=3 E:1588.2.1.1.1.8.5.1.4.1918=1
E:1588.2.1.1.1.8.5.1.5.1918="SEC-3138 Event: user inactivity warning,
USER user will expire in 2 days."
03:27:30.313334 IP 10.24.15.197.50000 > ldap.testsqa.com.SNMPtrap:
C="cm1" Trap(246) E:1588.2.1.1.1 10.24.15.197 enterpriseSpecific s=4
365800 S:18.1.3.0=10.24.15.197 E:1588.2.1.1.1.8.5.1.1.1919=1919
E:1588.2.1.1.1.8.5.1.2.1919="2020/12/26-02:53:09"
E:1588.2.1.1.1.8.5.1.3.1919=2 E:1588.2.1.1.1.8.5.1.4.1919=1
E:1588.2.1.1.1.8.5.1.5.1919="SEC-3139 Event: user expired USER
Extuser expired 3 days ago."
```

Locking out the inactive accounts enhances security by presenting lesser options for a brute force attacker to enter the system by making use of a `dormant SLX account` that could also be an administrator.

Forcing default users password change

This feature is introduced in SLX lately for hardening the security of management plane.

From the release of 20.3.1 configuration is available in SLX to force the password change of default users root, admin and user. By enabling this configuration when default users root, admin and user logs in for first time on SLX they are prompted to change the default password.

The default password for default admin and default user are `default config options` and the default password for `root user` is present in the factory settings of the device.

Note
The user is not allowed to login without changing the password upon first login for these users when this configuration is present.

The configuration to enable default user password change is as below.

```
SLX(config)# password-attributes force-default-password-change
```

Forcing default password change can prevent brute force attackers to enter the system and corrupt the file system via access to accounts like root. The password age of users other than root can be globally set from release 20.3.1 via the configuration command below.

```
SLX(config)# password-attributes max-password-age 100
```

In the above example global password age for all users other than root is `100` days. The age can be specified in the range `0 to 999` days where `0` disables password aging, where a password would never expire.

By having this configuration, a user after login is prompted to change the password after the specified number of days have expired, when the password was last modified.

GRUB Bootloader Password Protection

This feature is introduced in SLX lately for hardening the security of management plane.

Currently, there are no protections to access the `GRUB` boot loader. Any user with access to the console during boot may interrupt the boot sequence to enter `GRUB` without authenticating.

When a user with access to the console during boot interrupts the boot sequence to modify GRUB, the system prompts for and validate a credentials before entering GRUB menu entries other than default.

The following command configures grub bootloader password.

```
SLX(config-grub)# username root password fabos345
```

The GRUB credentials will be asked when accessing grub only, when the feature is enabled via configuring the above CLI.

Measured boot and Remote Attestation

This feature is introduced in SLX lately for hardening the security of management plane.

Measured boot and Remote Attestation are supported from 20.3.3 release.

Exploiting an embedded network device by planting the malware in one or more components of boot process is a type of security attack, which can go unnoticed as the malware may behave just like normal firmware.

Measured boot feature supports measuring the boot components and selected (custom) files during run time. Remote Attestation feature authenticates the hardware and software components (i.e., measurements from measured boot) to remote attestation server.

The following command is used to enable measured boot feature in SLX device,

```
SLX# measured-boot enable
```

Note
The device must be rebooted for the above CLI to take effect.

To support Remote attestation, user must setup (Keylime) registrar server, which is not in the scope of this document (please refer online Keylime server installation guide).

The following commands are used to configure Keylime agent that runs on the SLX device:

```
SLX(config)# remote-attestation
SLX(config-remote-attestation)# registrar-server <registrar-ipaddress>
```

To start Keylime agent on SLX device execute the below command.

```
SLX(config-remote-attestation)# agent-enable
```

Note
Refer Keylime server guides to start remote attestation using keylime-tenant utility.

Security Enhanced Linux (SE Linux)

Security-Enhanced Linux (SE Linux) is a Linux Kernel Module that enhances the security of SLXOS's underlying Linux OS. SE Linux works by providing security policies for access control at the operating system level. Support for Mandatory Access Control (MAC) is also available for use.

Security policies are a set of rules that implement access control restrictions for applications, processes, and files on the SLXOS's operating system. These rules are used by SE Linux to enhance security by preventing bypass of application security mechanism and enable containing the potential damages due to malicious or misbehaving applications.

Support for SE Linux is introduced in SLXOS version 20.4.1. As a part of this, MAC policy support for *SSHD* and *HTTPD* modules and their dependencies are added.

SE Linux has three modes of operation:

- In the *Disabled* mode, the operating system does not implement SE Linux policy and also does not label any persistent objects such as files. Not marking these persistent objects makes it harder to implement SE Linux in the future.
- In the *Permissive* mode, the operating system implements the SE Linux policy fully. All policy enforcement activities are logged. However, the policy is not enforced.
- In the *Enforcing* mode, the operating system implements the SE Linux policy completely including denying access, and activity logging.

SE Linux *Permissive* mode is enabled by default and cannot be changed.

Note

The last *1000 error log* entries will be saved in the `INFRA.txt` file within the support save logs.

Note

This feature is enabled on all platforms of SLXOS.

Use the **`show selinux status`**command to verify the current SE Linux status.

```
SLX # show selinux status
SE Linux status:                enabled
SE Linuxfs mount:               /sys/fs/selinux
SE Linux root directory:        /etc/selinux
Loaded policy name:             mls
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      31
```