Extreme NetIron FIPS Configuration Guide

06.3.00aa

9037139-00 Rev AA
July 2021

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, please 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:
www.extremenetworks.com/support/policies/software-licensing

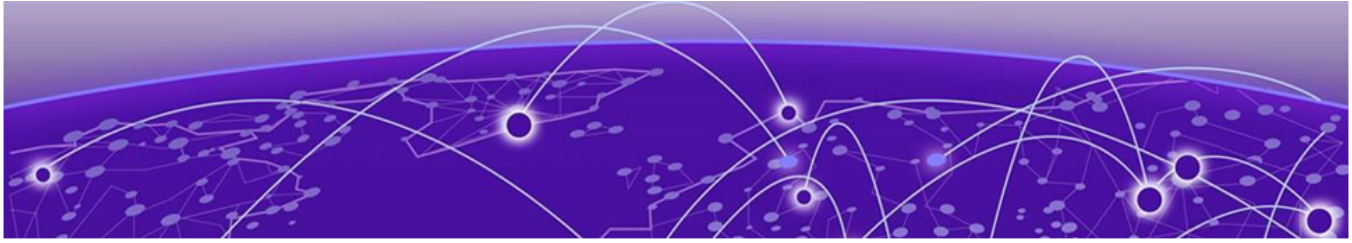


Table of Contents

Preface **6**
- Text Conventions 6
- Documentation and Training 7
- Getting Help 8
  - Subscribe to Service Notifications 8
- Providing Feedback 9

About This Document **10**
- Supported hardware and software 10
- FIPS-supported devices 10
- FIPS-supported interface modules 10
  - Extreme MLXe platform 11
  - Extreme CER platform 11
- What's new in this document 11

FIPS Support **12**
- FIPS overview 12
- How FIPS works 13
- Cryptographic Authentication of OSPFv2 and OSPFv3 15
- Configuring keychain support 15
- Configuring keychain for OSPFv2 virtual link 16
- Configuring Keychain for OSPFv3 virtual link 17

Upgrading and Downgrading Software on FIPS-enabled Devices **18**
- Upgrading FIPS-enabled devices 18
  - Image verification in FIPS mode 18
  - FIPS NetIron 06.3.00aa images for Extreme MLXe devices 19
  - Performing a basic upgrade 19
  - MACsec and software release upgrade 20
- Downgrading from FIPS mode to non-FIPS mode 21

FIPS Configuration **22**
- User roles in FIPS mode 22
- Commands disabled in FIPS mode 23
- Hidden files in FIPS mode 23
- Cryptographic algorithms in FIPS mode 23
  - Cryptographic algorithms on the management module 24
  - Cryptographic algorithms on the Extreme NetIron CER devices 25
  - Cryptographic algorithms on the BR-MLX-10GX4-IPSEC-M module 26
  - Cryptographic algorithms on the BR-MLX-10GX20-X2 and BR-MLX-1GX20-U10G-X2 modules 26
- SSH clients 26
- Usernames and SSH public key authentication 27

Implementation 27
Restrictions 27
Protocol changes in FIPS mode 28
BGP 29
HTTP 29
HTTPS 29
IKEv2/IPsec 30
IS-IS 31
L2 over IPsec 31
MACsec 31
MPLS 32
NTP 32
OpenFlow 33
OSPFv2 34
OSPFv3 34
PKI 35
Proprietary 2-way encryption algorithms 35
RADIUS 35
SCP 36
SNMP 37
SSHv2 38
Syslog 39
TACACS+ 40
Telnet 40
TFTP 40
VRRP 40
VRRP-E 40
Web Management 41
DRBG Health Test on IPsec MP 41
System reset and boot up in FIPS mode 42
Debugging in FIPS mode 42
Placing the device in FIPS mode 43
General steps to place the Extreme NetIron device in FIPS mode 43
Copying the signature files 43
Enabling FIPS mode 46
Zeroizing shared secrets and host keys 51
Configuring user authentication 54
Saving the configuration 56
Reloading the device 56
Performing a FIPS self-test 58
Modifying the FIPS policy 59
Disabling FIPS mode 60
Running FIPS self-test 61
Access to monitor mode 61
Accessing monitor mode from FIPS mode 62
Accessing monitor mode in the event of continuous failure 63
Debugging in monitor mode 63
Returning to FIPS mode from monitor mode 63

Appendix: SP800-90A DRBG Implementation 64

DRBG support information....64

Preface

This section describes the text conventions used in this document, where you can find additional information, and how you can provide feedback to us.

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
Archived Documentation (for earlier versions and legacy products)
Release Notes
Hardware/software compatibility matrices for Campus and Edge products
Supported transceivers and cables for Data Center products
Other resources, like 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/.

Getting Help

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 Service Notifications

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

1. Go to www.extremenetworks.com/support/service-notification-form.
2. Complete the form (all fields are required).
3. Select the products for which you would like to receive notifications.

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

4. Select **Submit**.

Providing Feedback

The Information Development team at Extreme Networks has made every effort to ensure the accuracy and completeness of this document. We are always striving to improve our documentation and help you work better, 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 in the document.
- Broken links or usability issues.

If you would like to provide feedback, you can do so in three ways:

- In a web browser, select the feedback icon and complete the online feedback form.
- 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

Supported hardware and software on page 10
FIPS-supported devices on page 10
FIPS-supported interface modules on page 10
What's new in this document on page 11

Supported hardware and software

The following hardware platforms are supported by FIPS:

- Extreme NetIron CER 2000-4X-RT Series
- Extreme MLXe series (MLXe-4) with management module (BR-MLX-MR2-X)

FIPS-supported devices

Federal Information Processing Standards (FIPS) is vendor-ready for the following devices:

- Extreme MLXe-4 with management module
- Extreme MLXe with management module BR-MLX-MR2-X: 1666 MHz Power PC processor 7448 (version 8004/0202) 166 MHZ bus
- Extreme CER (including 4X-RT models): 800 MHz Power PC processor 8544E (version 8021/0022) 400 MHz bus

Note

Refer to the release notes for the software version running on the device to verify that software is certified for FIPS and Common Criteria.

Table 4: Devices that support FIPS

Extreme NetIron XMR	Extreme MLX Series	Extreme MLXe Series with management module (MLXe-4)	Extreme NetIron CER 2000 Series (4X-RT models only)
No	No	Yes	Yes

FIPS-supported interface modules

FIPS is vendor-ready for the following interface modules:

- BR-MLX-10Gx20-X2
- BR-MLX-1GX20-U10G-X2

- BR-MLX-10GX4-IPSEC-M

Extreme MLXe platform

The following Extreme MLXe management modules are supported for certification:

- BR-MLX-MR2-X

The following Extreme MLXe chassis bundles are supported for certification:

- BR-MLXE-4-MR2-X-AC

The following Extreme MLXe switch fabric modules are supported for certification:

- NI-X-4-HSF

Note
For more information about the modules and their descriptions, refer to the specific Extreme NetIron hardware installation guides.

Extreme CER platform

The following Extreme CES chassis bundles are supported for certification:

- BR-CER-2024C-4X-AC
- BR-CER-2024F-4X-AC

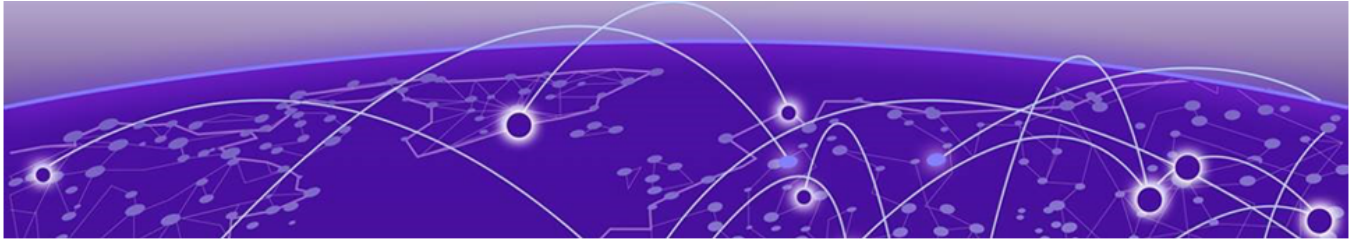
Note
For more information about the modules and their descriptions, refer to the specific Extreme NetIron hardware installation guides.

What's new in this document

The new features for this release are:

- OSPFv2: RFC 7474, OSPFv2 HMAC-SHA Crypotographic Authentication
- OSPFv3: RFC 7166, Supporting Authentication Trailer for OSPFv3
- Key chain management infrastructure to support OSPFv2/OSPFv3 Authentication changes
- OpenSSL upgrade to 1.0.2p for both MP and LP

For the complete list of supported features and the summary of enhancements and configuration notes for this release, refer to the NetIron 6.3.00a Release Notes.



FIPS Support

FIPS overview on page 12
How FIPS works on page 13
Cryptographic Authentication of OSPFv2 and OSPFv3 on page 15
Configuring keychain support on page 15
Configuring keychain for OSPFv2 virtual link on page 16
Configuring Keychain for OSPFv3 virtual link on page 17

FIPS overview

An Extreme device in Federal Information Processing Standards (FIPS) mode is compliant with the standards established by the United States government and the National Institute of Standards and Technology (NIST).

Note
Not all software releases support FIPS. Refer to the release notes to verify if the software you are running supports FIPS.

The FIPS Publication 140-2 is a technical standard and worldwide de-facto standard for the implementation of cryptographic modules. The FIPS Publication 140-2 contains security standards developed by the United States government and the National Institute of Standards and Technology (NIST) for use by all non-military government agencies and by government contractors. Due to their importance within the security industry, these standards form a baseline for many security requirements.

In FIPS mode, the network processing occurs in the kernel and in privileged daemons.

You can configure the Extreme device to run in FIPS mode to ensure that the device is operating according to the standards stated in FIPS Publication 140-2.

An Extreme device is FIPS 140-2-compliant when the following requirements have been met:

- Tamper-evident security seals labels are applied to the device according to the instructions included in the tamper-resistant accessory kit. The accessory kit must be purchased separately.
- The device software is placed in FIPS mode with the FIPS security policy applied.

Note
Tamper-evident security seals must be applied to the product. For details on how to place the tamper-evident security seals, refer to the platform-specific *FIPS Security Seal Procedures* document available on www.extremenetworks.com.

Note
Once FIPS mode is enabled on the system, even if the mode is disabled at a later time, a firmware integrity test will always be carried out on the device at image copy time.

How FIPS works

You place a device in FIPS mode by entering the **fips enable** command on the management station while the station is connected to the device console port with a serial cable. After you enter the **fips enable** command, the device is administratively in FIPS mode and by default runs in strict FIPS-compliant mode upon reload.

In addition, you can configure an optional set of FIPS policy commands, and then use the **fips zeroize all** command to zero out the shared secrets used by various networking protocols, including the host access passwords, and the SSH and HTTPS host and client keys based on the configured FIPS security policy. After you issue the **fips zeroize all** command, use the **write memory** command, and then place the device in FIPS operational mode by reloading the device.

The default FIPS policy is for the system to run in a strict mode that fully supports FIPS 140-2 specifications. However, the device allows you the flexibility to configure a modified FIPS policy according to your network requirements. Refer to Modifying the FIPS policy.

Note
A FIPS policy that varies from the default policy weakens the intent of the FIPS 140-2 specifications; when implemented, the device is not operating in full compliance with these specifications.

The default FIPS approved mode enables the following actions for strict FIPS compliance:

- The SCP
- HTTPS TLS v1.1 and TLS v1.2

Note
Use of the **debug** command violates the Security Policy of the module and it deems the module non-compliant in the FIPS mode.

The default FIPS approved mode disables the following actions for strict FIPS compliance:

- Telnet access including the **telnet server** command.
- AAA authentication for the console using enable aaa console command is temporarily disabled to allow console access to configure SSH parameters. This command can be enabled after SSH is confirmed operational.
- The **ip ssh scp disable** command.
- The TFTP access.
- SNMPv3 access to Critical Security Parameters (CSP) specific MIB objects.
- Access to all commands that allows debugging memory content within the monitor mode.
- HTTP access including the **web-management http** command.
- The HTTPS SSL 3.0 access.

- The **web-management allow-no-password** command.
- TACACS which is a UDP based protocol

The default FIPS approved mode clears the following actions for strict FIPS compliance:

- Protocol shared secret and host passwords
- HTTPS RSA host keys and certificate

The FIPS mode zeroizes shared secrets and passwords.

Note
Users are expected to explicitly enter the **fips zeroize all** command to zeroize shared secrets, passwords, and host keys before placing the device in FIPS mode.

Note
Note that Group 14, Group 19, and Group 20 parameters are allowed in IKEv2/IPsec protocols in FIPS mode.

The HTTPS server allows the following ciphers:

- CIPHERSUITE_RSA_WITH_AES_128_CBC_SHA
- CIPHERSUITE_RSA_WITH_AES_256_CBC_SHA
- CIPHERSUITE_DHE_RSA_WITH_AES_128_CBC_SHA
- CIPHERSUITE_DHE_RSA_WITH_AES_256_CBC_SHA
- CIPHERSUITE_RSA_WITH_AES_128_CBC_SHA256
- CIPHERSUITE_RSA_WITH_AES_256_CBC_SHA256
- CIPHERSUITE_DHE_RSA_WITH_AES_128_CBC_SHA256
- CIPHERSUITE_DHE_RSA_WITH_AES_256_CBC_SHA256

After defining the FIPS policy, save the configuration, and restart the device. While the device is restarting, several tests are run to ensure the device is FIPS-compliant.

Some of these tests include several FIPS self-tests such as Known Answer Tests (KATs) and conditional tests that are run to ensure that the cryptographic engine is FIPS-compliant.

After these tests are run successfully, the device reloads and is operationally in FIPS mode. All the optional FIPS policy commands are provided to perform various non-approved FIPS operations when FIPS is enabled. Note that if any of these policy commands are configured, then the module does not operate in the approved FIPS mode.

Note
Execution of the **self-test** command in FIPS operational or administration modes may result in the device restarting as per the FIPS criteria if any of the algorithm self-tests fails.

Cryptographic Authentication of OSPFv2 and OSPFv3

You can configure the authentication used for a specific routing protocol by specifying the key and cryptographic algorithm. This feature supports FIPS 140-2 Compliant Cryptographic Authentication for OSPFv2 and OSPFv3 using HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512.

In addition, following features are supported.

- Protection against replay attack on routing Protocols.
- Protection against routing tables update by unauthorized peer for OSPFv2 and OSPFv3.
- Key rollover without bringing down the adjacency.
- Global configuration of keys and other relevant parameters.

Configuring keychain support

Key chains are sequences of keys. Users can configure key chains and can use keys with features that secure communications with other devices by using key-based authentication and optionally perform periodic key rotations within the chain. The configured keychain can be used for any of the supported routing protocols.

Procedure

1. Enter global configuration mode.

   ```
   device# keychain ospfv1kc
   ```

2. Enter keychain configuration mode. Up to 128 keychains can be configured. Valid name length is from 4 characters through 32 characters. No special characters are allowed, except for the underscore and hyphen.

   ```
   device(config)# keychain  keychain1
   device(config-keychain1)#
   ```

3. Configure the acceptance tolerance for the key.

   ```
   device(config-keychain1)# accept-tolerance 100
   ```

   The range of valid values is from 0 through 600 seconds.

4. Enter key configuration mode. The range of valid values is from 1 through 65535.

   ```
   device(config-keychain1)# key 100
   device(config-key-100)#
   ```

   The keychain configurations contain only the default values until modified by other commands.

5. Set the key-string using following command.

   ```
   device(config-key-100)# key-string Mystring1
   ```

   The valid values are 0 and 7. configuration of key-string is required before key can be used.

6. Configure the acceptance lifetime for the key. You can specify either the end-time or state infinite. You also have an option to use local or GMT, duration, or infinity.

   ```
   device(config-key-100)# accept-lifetime local start-time 13:40:40|12/07/2018 end-time
   11:40:40|14/07/2018
   ```

   Default value for this parameter is 0, which means that the key is not active until the lifetime is configured. By default this command is not set.

7. Set the hash algorithm for the key.

```
device(config-key-100)# key-algorithm ?
```

The valid algorithms are HMAC-SHA-1(1), HMAC-SHA-256(2), HMAC-SHA-384(3), and HMAC-SHA-512(4). The default algorithm is HMAC-SHA-256.

Example

The following is an example of configuring a single keychain and key.

```
device# configure terminal
device(config)# keychain  keychain1
device(config-keychain1)# accept-tolerance 500
device(config-keychain1)# key 100
device(config-key-100)# key-string Mystring1
device(config--key-100)# accept-lifetime local start-time 22:57:40|07/04/2018 end-time
23:59:59|12/04/2018
device(config--key-100)# do show running-config keychain keychain1
keychain keychain1
 accept-tolerance 500
 key 1
  key-string $9$XutLBELmbQ765dsLycIP/A==
  accept-lifetime gmt start-time 23:00:50|07/04/2018 end-time 23:59:59|12/04/2018
  key-algorithm HMAC-SHA-256
 !
device(config-key-100)# exit
device(config)#
```

Configuring keychain for OSPFv2 virtual link

Key chains are sequences of keys (shared secrets). Users can configure key chains and can use keys with features that secure communications with other devices by using key-based authentication and optionally perform periodic key rotations within the chain. For OSPFv2 the new authentication feature can be applied on an interface and a virtual link.

Procedure

Configuring OSPFv2 Authentication feature:

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter the **router ospf** command to enter OSPF router configuration mode and enable OSPFv2 on the device.
3. Enter the **area** command to assign an OSPFv2 area ID.

```
device(config-router-ospf)# area 0
```

4. Enter the **area** command to assign a second OSPFv2 area ID.

```
device(config-router-ospf)# area 1
```

5. Enter the **interface** command and the ID of the OSPFv2 device at the remote end of the virtual link to configure the virtual link endpoint.

   Configuring OSPFv2 Authentication feature under Interface:

```
device(config)# interface ethernet 1/1
device(config)# enable
device(config)# ip ospf authentication key-chain keychain2
device(config)# ip address 53.54.43.54/24
device(config)# geg-default neg-off
```

6. Enter the **area virtual-link** command and the ID of the OSPFv2 device at the remote end of the virtual link to configure the virtual link endpoint.

```
device(config-router-ospf)# area 1 virtual-link 3.3.3.3
device(config-router-ospf)# area 1 virtual-link 3.3.3.3 authentication key-chain
keychain1
```

Configuring Keychain for OSPFv3 virtual link

Key chains are sequences of keys (shared secrets). Users can configure key chains and can use keys with features that secure communications with other devices by using key-based authentication and optionally perform periodic key rotations within the chain. For OSPFv3 the new authentication feature can be applied on an interface and a virtual link.

Procedure

Configuring OSPFv3 Authentication feature:

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter the **router ospf** command to enter OSPF router configuration mode and enable OSPFv3 on the device.
3. Enter the **area** command to assign an OSPFv3 area ID.

```
device(config-ipv6-router-ospf)# area 0
```

4. Enter the **area** command to assign an OSPFv3 area ID.

```
device(config-ipv6-router-ospf)# area 1
```

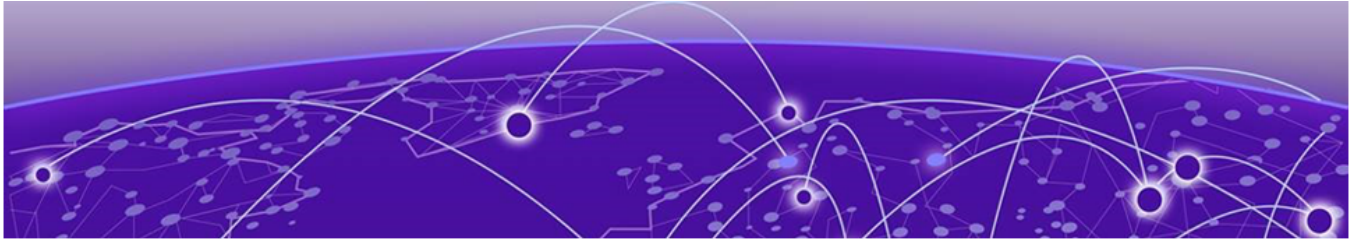
5. Enter the **interface** command and the ID of the OSPFv3 device at the remote end of the virtual link to configure the virtual link endpoint.

   Configuring OSPFv3 Authentication feature under Interface:

```
device(config)# interface ethernet 1/1
device(config)# enable
device(config-ipv6-router-ospf)# ipv6 ospf authentication key-chain keychain2
device(config-ipv6-router-ospf)# ipv6 address 20.20::54/64
device(config-ipv6-router-ospf)# gig-default neg-off
```

6. Enter the **area virtual-link** command and the ID of the OSPFv3 device at the remote end of the virtual link to configure the virtual link endpoint.

```
device(config-ipv6-router-ospf)# area 1 virtual-link 3.3.3.3
device(config-ipv6-router-ospf)# area 1 virtual-link 3.3.3.3 authentication key-chain
keychain1
```



Upgrading and Downgrading Software on FIPS-enabled Devices

Upgrading FIPS-enabled devices on page 18
Downgrading from FIPS mode to non-FIPS mode on page 21

Upgrading FIPS-enabled devices

FIPS 140-2 compliance is a combination of implemented hardware procedures and the activation of a software-based security policy.

Note
Although commands to alter the FIPS security policy exist, altering the default FIPS security policy is not recommended.

Note
After enabling FIPS mode on your device, you cannot disable it without losing the device configuration. To disable FIPS mode, it is recommended that you contact Extreme Technical Support and perform the procedure under qualified guidance.

Image verification in FIPS mode

Upgrading from non-SHA256 signatures to SHA256 signature packages requires two upgrade cycles to update the signature files to SHA256 signatures for LP-auto-upgrade to use the SHA256 signatures for manifest file signature check.

Note
Refer to the latest version of the Extreme NetIron Release Notes for the list of images.

When upgrading from a release that does not support SHA256 signatures to a release that does, upgrade twice to the same release as follows. First upgrade to the release that supports SHA256 signatures. Reload the device. Then upgrade again to the same release that supports SHA256 signatures, and reload the device again. This ensures that the device will have the SHA256 signatures on the device.

Note
LP auto-upgrade is not supported in FIPS mode.

FIPS NetIron 06.3.00aa images for Extreme MLXe devices

Note

Once a device has been cryptographically validated for FIPS (as indicated in the **`fips show`** output), signature verification of images is always done at the time of uploading the images to the device. To un-validate a cryptographically validated FIPS module, contact Extreme technical support.

Table 5: Required images for a basic upgrade to NetIron 06.3.00aa

Image description	Image name	Signature name for upgrade from devices running (legacy) NetIron 06.3.00aa and earlier code using DSA1024/SHA1 signatures	RSA2048/SHA256 bit signatures file name
Combined application image for management modules	xm06300aa.bin	xmr06300aa.sig xmlp06300aa.sig	xm06300aa.sha256 xmlp06300aa.sha256
Monitor image for management modules	xmb06200.bin	xmb06200.sig	xmb06200.sha256
Monitor image for interface modules	xmlb06200.bin	xmlb06200.sig	xmlb06200.sha256
Boot image for management modules	xmprm05900.bin	xmprm05900.sig	xmprm05900.sha256
Boot image for interface modules	xmlprm05900.bin	xmlprm05900.sig	xmlprm05900.sha256
Combined FPGA image for interface modules	lpfpga06300aa.bin	lpfpga06300aa.sig	lpfpga06300aa.sha256

Performing a basic upgrade

The overall procedure for a basic upgrade involves copying only the new application, boot, monitor, and combined FPGA images.

There are two ways to perform an upgrade to FIPS-enabled devices:

- Using Secure Copy (SCP). For more information about SCP, refer to the Extreme NetIron configuration guides.

- Using a TFTP server. To upgrade using TFTP at the Privileged EXEC level of the CLI (**fips policy allow tftp-access** is enabled), you must first enter the command in global configuration mode:

```
device(config)# fips policy allow tftp-access
```

Note

- If the device is in FIPS mode, use the **fips policy allow tftp-access** command. If the device is not in FIPS mode, TFTP is allowed.
- Once FIPS mode is enabled on the system, even if the mode is disabled at a later time, the firmware integrity test will always be carried out on the device at image copy time. The RSA2048-SHA256-based signature firmware integrity test is run during image installation time and during image reload time when the device has been administratively enabled for FIPS. The test is run on MP and LP images at image reload time, when the device is in the FIPS mode. This test is in addition to the CRC-16 test that is run by the device during image reload time. Both the tests should pass for the device to reload successfully.
- Before upgrading the image, if the device does not have the correct signature files on the device, and the target image is the same as the current image on the device, then we need to run the **force-sync-standby** command. Note that you should run the command after the image upgrade and before the device reload. The specific signatures files may not be available if they were removed or not installed before the upgrade attempt, and the image being upgraded to is the same as the one which is on the device prior to the upgrade. For this reason, it is preferable to use simplified upgrade to allow for the correct signatures to be copied simultaneously with the image.

MACsec and software release upgrade

If the device has MACsec configuration (for example, using the **dot1x-mka-enable** command) and you are upgrading the software, the bypass test (also known as the FIPS Integrity Qualification test) is automatically executed when the device is in FIPS mode. This test generates the HMAC values based on the available MACsec configurations. The output of the **show running** command displays the **fips bypass-test macsec config-integrity** command along with the HMAC value.

```
device# show running
!
fips enable
fips bypass-test macsec config-integrity "8f67c6019f82b1657fc704c4d8e78f37c9c6aa73"
!
```

Note

The **fips bypass-test macsec config-integrity** command is an auto-generated command. You cannot execute this command manually in the CLI.

Downgrading from FIPS mode to non-FIPS mode

About This Task

Downgrading from FIPS mode to non-FIPS mode clears all shared secrets, host passwords, SSH and HTTPS host keys and HTTPS certificates.

Note

Once FIPS mode is enabled on the system, even if the mode is disabled at a later time, the firmware integrity test will always be carried out on the device at image copy time. The RSA2048-SHA256-based signature firmware integrity test is run during image installation time and during image reload time when the device has been administratively enabled for FIPS. The test is run on MP and LP images at image reload time, when the device is in the FIPS mode. This test is in addition to the CRC-16 test that is run by the device during image reload time. Both the tests should pass for the device to reload successfully.

Note

In FIPS mode, do not attempt to downgrade to a release that does not support SHA256 signatures. Generally, releases prior to Extreme NetIron 5.6.00c (excluding 5.6.00aa) do not support SHA256 signatures. In FIPS mode, downgrading to release that does not support SHA256 signatures is not supported.

Note

All shared-secret passwords (including any MD5 passwords) are lost when downgrading from a FIPS environment to a non-FIPS environment.

To place a device in non-FIPS mode and then use TFTP or SCP to download and initialize an older image, complete the following steps.

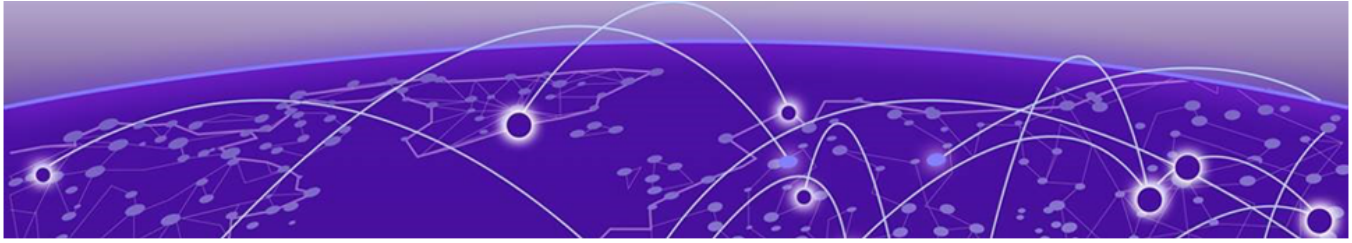
Procedure

1. Log in to the device by entering your username and password.
2. Disable FIPS by entering the **`no fips enable`** or **`no fips enable`** command at the prompt.
3. Regenerate SSH host keys or other shared secrets as needed for access after reload.
4. To replace the startup configuration with the **`no fips enable`** configuration, enter the **`write memory`** command.

```
device# write memory
```

5. Reload the configuration by entering the **`reload`** command.

What to Do Next



FIPS Configuration

User roles in FIPS mode on page 22
Commands disabled in FIPS mode on page 23
Hidden files in FIPS mode on page 23
Cryptographic algorithms in FIPS mode on page 23
SSH clients on page 26
Usernames and SSH public key authentication on page 27
Protocol changes in FIPS mode on page 28
DRBG Health Test on IPsec MP on page 41
System reset and boot up in FIPS mode on page 42
Debugging in FIPS mode on page 42
Placing the device in FIPS mode on page 43
Disabling FIPS mode on page 60
Running FIPS self-test on page 61
Access to monitor mode on page 61

User roles in FIPS mode

Configuring FIPS mode on the Extreme devices complies with the standards established by the United States government and the National Institute of Standards and Technology (NIST).

An Extreme device in FIPS mode supports three user roles:

- Crypto-officer role: The Crypto-officer role on the device in FIPS mode is equivalent to the administrator role, or the super-user role in non-FIPS mode.
- Port Configuration Administrator role: The Port Configuration Administrator on the device in FIPS mode is equivalent to the port configuration user in non-FIPS mode and has write access to the interface configuration mode only.
- User role: The User role on the device in FIPS mode has read-only privileges and no configuration mode access.

Concurrent operators are supported, but no limit is enforced. The number of concurrent users is only limited by the system resources.

In addition to the user roles, the following roles support specific protocols:

- MACsec Peer role: The MACsec Peer role is available on the device. It allows MACsec Key Agreement (MKA) protocol sessions to be established with a remote peer based on the MACsec configuration

on the Extreme NetIron device. Once the Secure Association Keys (SAK) are obtained, the MACsec peer role will install the keys on the PHY and start MACsec communication with the peer.

- IKEv2/ IPsec Peer role: The IKEv2 Peer role is available on the IPsec-supported line cards. It allows Internet Key Exchange (IKE) and IPsec sessions to be established with a remote peer based on the IPsec configuration on the Extreme NetIron device
- NTP Peer role: This role performs the Network Time Protocol (NTP) operation.

Commands disabled in FIPS mode

The device in FIPS mode does not support the following commands:

- **enable password-display**
- **enable strict-password-enforcement**

Note
Strict password enforcement is enabled by default. The password must be at least eight characters long.

- **web-management allow-no-password**
- **telnet server**
- **ip ssh scp disable**
- **ip ssh key-authentication no**
- **ip ssh permit-empty-password no**
- **web-management http**
- **enable password-display**

A device in FIPS mode does not support the following TFTP commands:

- **copy tftp flash** *ip*
- **boot system tftp** *ip file*
- **ip ssh pub-key-file tftp** *ip* {*file* | *pubkey*}
- **ip ssl certificate-data-file tftp** *ip file*
- **ip ssl private-key file tftp** *tftp file*

Hidden files in FIPS mode

Hidden files are not displayed when the device is in FIPS mode. Hidden files are displayed only when the device is in non-FIPS mode.

Cryptographic algorithms in FIPS mode

The device in FIPS mode supports the following FIPS 140-2-approved cryptographic algorithms:

- Cryptographic Algorithms on the Management module
- Cryptographic Algorithms on the Extreme NetIron CER devices
- Cryptographic Algorithms on the BR-MLX-10GX4-IPSEC-M module
- Cryptographic Algorithms on the BR-MLX-10GX20-M and BR-MLX-10GX20-X2 modules

Allowed exceptions include:

- RSA Key Wrapping
- Message Digest 5 (MD5)
- Hash Message Authentication Codes - Message Digest 5 (HMAC-MD5) as used in RADIUS
- Non-Deterministic Random Number Generator (NDRNG)

The device in FIPS mode does not support the following cryptographic algorithms:

- DES
- 3-DES
- RSA 1024-bit key size
- SSH key exchange algorithm (diffie-hellman-group1-sha1)
- SNMPv1
- SNMPv2C
- SNMPv3 in noAuthNoPriv and authNoPriv security mode
- HMAC-SHA1-96

Cryptographic algorithms on the management module

The management module in FIPS mode supports the following FIPS 140-2-approved cryptographic algorithms:

- Advanced Encryption Algorithm (AES) including AES-CBC, AES-CTR, and AES-CFB
- AES Key Wrap (KW) RFC 3394
- Cipher-based MAC (CMAC) with AES 128
- Secure Hash Algorithm (SHA) (including all SHA variants the module supports: SHA-1, SHA-256, and SHA-384)
- Key-Based Key Derivation Functions (KBKDF SP800-108)
- Keyed-Hash Message Authentication Code (HMAC-SHA1, HMAC-SHA256)
- Counter-based Deterministic Random Bit Generator (DRBG)
- Rivest Shamir Adleman (RSA) signature algorithm including RSA2, FIPS 186-4 KeyGen, SigGen, SigVer
- Elliptic Curve Digital Signature Algorithm (ECDSA) FIPS 186-4 KeyGen, SigGen, SigVer
- TLS 1.1 and TLS 1.2 KDF SP800-135
- SSH Key exchange algorithm diffie-hellman-group-exchange-sha256
- SNMPv3 (in authPriv security mode) KDF SP800-135
- SSHv2 Key Derivation Function (KDF)

Allowed exceptions include:

- RSA Key Wrapping
- Message Digest 5 (MD5)
- Hash Message Authentication Codes - HMAC-MD5
- Non-Deterministic Random Number Generator (NDRNG)

The device in FIPS mode does not support the following cryptographic algorithms:

- DES
- 3-DES
- HMAC-SHA1-96
- RSA 1024-bit key size
- SSH key exchange algorithm (diffie-hellman-group1-sha1)
- SNMPv1
- SNMPv2C
- SNMPv3 in noAuthNoPriv and authNoPriv security mode

Cryptographic algorithms on the Extreme NetIron CER devices

The Extreme NetIron CER devices in FIPS mode support the following FIPS 140-2-approved cryptographic algorithms:

- Advanced Encryption Algorithm (AES) including AES-CBC, AES-CTR, and AES-CFB
- Secure Hash Algorithm (SHA) (including all SHA variants the module supports: SHA-1, SHA-256, and SHA-384)
- Keyed-Hash Message Authentication Code (HMAC-SHA1, HMAC-SHA256)
- Counter-based Deterministic Random Bit Generator (DRBG)
- Rivest Shamir Adleman (RSA) signature algorithm including RSA2, FIPS 186-4 KeyGen, SigGen, SigVer
- TLS 1.1 and TLS 1.2 KDF SP800-135
- SSH Key exchange algorithm diffie-hellman-group-exchange-sha256
- SNMPv3 (in authPriv security mode) KDF SP800-135
- SSHv2 Key Derivation Function (KDF)
- Allowed exceptions include:
  - RSA Key Wrapping
  - Message Digest 5 (MD5)
  - Hash Message Authentication Codes - HMAC-MD5
  - Non-Deterministic Random Number Generator (NDRNG)

  The device in FIPS mode does not support the following cryptographic algorithms:

  - DES
  - 3-DES
  - HMAC-SHA1-96
  - RSA 1024-bit key size
  - SSH key exchange algorithm (diffie-hellman-group1-sha1)
  - SNMPv1
  - SNMPv2C
  - SNMPv3 in noAuthNoPriv and authNoPriv security mode

Cryptographic algorithms on the BR-MLX-10GX4-IPSEC-M module

The Extreme NetIron BR-MLX-10GX4-IPSEC-M module in FIPS mode supports the following FIPS 140-2-approved cryptographic algorithms:

- IKEv2 KDF SP800-135
- KAS ECC SP800-56A
- KAS FFC SP800-56A
- DRBG SP800-90A
- AES (AES-256-ECB)
- GCM (SP800-38D)

Algorithms running on the onboard security engine:

- AES (AES-128-CBC and AES-256-CBC)
- SHA (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)
- Elliptical Curve Diffie-Hellman (ECDH)
- ECDSA

Algorithms running on the IPsec FPGA:

- AES (AES-128-GCM)
- AES (AES-256-GCM)

Algorithms running on the MACsec PHY:

- Clarity MACsec

Cryptographic algorithms on the BR-MLX-10GX20-X2 and BR-MLX-1GX20-U10G-X2 modules

The Extreme NetIron BR-MLX-10GX20-X2 and BR-MLX-1GX20-U10G-X2 modules have an onboard MACsec PHY chip that supports the following FIPS 140-2-approved cryptographic algorithms:

- AES (AES-128-GCM)

SSH clients

SSH clients must be FIPS 186-4-compliant. You can use the OpenSSH-based client that is developed by Extreme to be FIPS 186-4-compliant.

When doing SCP of a file from MLX device using OpenSSH, please do the following:

- Use following command to do th SCP transfer on the client. The default for channel bandwidth is 8192kbits/sec. This prevents SCP to stall. **`scp-vvv-|8192 <username>@<MLX IP>:slot1|2|flash:<filename>`**
- Add **`ServerAliveInterval`** and **`ServerAliveCountMax`** options to **`ssh_config`** to prevent SCP termination due to broken pipe error, which can occur due to server inactivity. The error can also occur due to duplication in MLX IP addresses.

Usernames and SSH public key authentication

The device stores or uses the username that is provided by the SSH client when public-key authentication is used. Therefore, the username is mentioned in the login and logout syslogs.

The devices save the username from the public-key authentication request. The username is used in the login and logout syslogs. When FIPS mode is operational, the device uses the username to match against the username attached to the SSH client public key stored on the device. If the two usernames do not match, the authentication request is denied.

Implementation

The client public key file format allows for a username to be provided in the "Subject" field the SSH2 public key. Additional private headers can be used. The privilege level can take three values : 0 READ-WRITE/ADMINISTRATOR, 4 PORT-CONFIG, and 5 READ-ONLY. The following public key example shows the two headers that are used by the device. No continuation lines are allowed in the file for these headers.

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "rsa-key-20121206"
Subject: brcd
x-extreme-privilege-level: 0
AAAAB3NzaC1yc2EAAAABJQAAAQEAkwiApY1x4T/DHII5JzR2OgqcF5vjlubNcvSE
UjkGmiRBDSOicjxS0ZLm1b2xFpVzw8XxSSy8cxvntfs5ortOt80QzynqgL+H2zJa
Lb4Qbu6/1vakJbPb/VUJE66Zezh0c8mze6zTbiP4iQ/Wn2lxpSmlS5cdowmFlZ7B
97xcagJIBl+7JKuvj8P+85ESUf2/pcrogqx7gdr1IpP2nev5s4xwCWFGtr2R/yMF
Q9h0xLcc4A7vLTDuY/h1GzLdICgtNYdqpUhpw+w0DkTKbQuDPd0gkwHkoFwg85lE
4VCDevdC/DeOCNJjNp9NbVD+SW6uL4NymmV7/i0YbPyl3gTESQ==
---- END SSH2 PUBLIC KEY ----
```

After decoding the base64 encoded public keys to binary format, a SHA256 hash of the binary format key is created. This hash is saved to memory. Verify that the hash is unique across the hashes of client public keys that have already been parsed. Additionally, non-empty usernames are also verified to be unique across the usernames already parsed in the public key. Access is denied if the usernames are mismatched.

The username has the following restrictions:

- The username cannot contain control characters, spaces, ", ?, |, or characters above ASCII code 0x7F.
- The username must be less than or equal to 48 characters.
- The username must be specified with the public key for that key to allow access. The user must specify a non-empty username in the login request.

Restrictions

No EXEC authorization through the AAA server is available because the privilege level is obtained from the public key file private header field (x-extreme-privilege-level) as shown in the public key example in Implementation.

Protocol changes in FIPS mode

The following table lists the protocols that undergo changes while the device is in FIPS mode with the default policy applied.

Table 6: Protocol changes

Protocols/Algorithms	Supported in FIPS mode	Supported in Non-FIPS mode	For more information on individual protocol changes, refer to the following sections
BGP	No	Yes	BGP
HTTP	No	Yes	HTTP
HTTPS	Yes	Yes	HTTPS
IPsec	Yes, with limitations	Yes	IKEv2/ IPsec
IS-IS	No	Yes	IS-IS
L2overIPsec	Yes	Yes	L2overIPsec
MACsec	Yes	Yes	MACsec
MPLS	No	Yes	MPLS
NTP	Yes, with limitations	Yes	NTP
OpenFlow	Yes	Yes	OpenFlow
OSPFv2	Yes	Yes	OSPFv2
OSPFv3	Yes	Yes	OSPFv3
PKI	Yes	Yes	PKI
Proprietary 2-way encryption algorithms	No	Yes	Proprietary 2-way encryption algorithms
RADIUS	Yes, with limitations	Yes	RADIUS
SCP	Yes	Yes	SCP
SNMP	Yes, with limitations	Yes	SNMP
SSHv2	Yes, with limitations	Yes	SSHv2
Syslog	Yes	Yes	Syslog
Telnet	No	Yes	Telnet
TACACS+	Yes, with limitations	Yes	TACAS+
TFTP	No	Yes	TFTP
VRRP	Yes	Yes	VRRP

Table 6: Protocol changes (continued)

Protocols/Algorithms	Supported in FIPS mode	Supported in Non-FIPS mode	For more information on individual protocol changes, refer to the following sections
VRRPe	Yes	Yes	VRRPe
Web Management	Yes	Yes	Web Authentication

Note
For more information on RADIUS authentication commands, refer to the *Extreme NetIron Command Reference* and the *Extreme NetIron Management Configuration Guide*.

BGP

Border Gateway Protocol (BGP) allows peer-to-peer authentication or client-to-server authentication.

To authorize an authentication, use a command such as the following to configure shared secret keys for BGP:

```
device(config-bgp-router)# neighbor 192.168.1.2 password P@$$w0rd
```

For more information on BGP authentication commands, refer to the *Extreme NetIron Routing Configuration Guide*.

HTTP

HTTP is not supported on the device in FIPS mode.

The **web-management http** command is disabled if it is included in the device's configuration. When the HTTP server is enabled because the **web-management http** command has been configured, the system removes the command from the configuration and the device displays the following message:

```
FIPS Compliance: HTTP service will been disabled
```

HTTPS continues to be enabled in FIPS mode and the configuration changes the **web-management http** command to the **web-management https** command.

HTTPS

The following HTTPS operations are affected in the FIPS approved mode:

- The **web-management https** command is maintained and offers equivalent functionality to the disabled **web-management http** command. Note that in addition to port 443, port 280 is also open for access by HP ProCurve Manager. You can disable this port using the **no web-management hp-top-tools** command.
- The **web-management allow-no-password** command is disabled.
- The **ip ssl certificate-data-file tftp** command is disabled when TFTP operation is disabled in FIPS mode. SCP supports functionality of the command. Refer to SCP.

- The **ip ssl private-key-file tftp** command is disabled when TFTP operation is disabled in FIPS mode. SCP supports the functionality of this command. Refer to SCP.
- SSL version 3 and earlier versions are disabled and TLS 1.1 or later versions are enabled.
- RC4 in TLS is disabled.
- RSA (PKCS #1 v2.1), or ECDSA (ANSI X9.62) for signature generation and verification.

TLS implementation in NetIron devices

By default, all TLS versions are supported on devices that act as an HTTPS server.

For devices that act as an SSL server or HTTPS server, the default connection is with TLS 1.2. For devices that act as an SSL client or syslog, OpenFlow, or secure AAA client, during session negotiation, the TLS version is decided based on the server support.

You can configure the minimum TLS version on NetIron devices using the **ip ssl server min-version** { 1 | 2 } command.

The following cipher suites are allowed in FIPS mode:

- CIPHERSUITE_RSA_WITH_AES_128_CBC_SHA
- CIPHERSUITE_RSA_WITH_AES_256_CBC_SHA
- CIPHERSUITE_DHE_RSA_WITH_AES_128_CBC_SHA
- CIPHERSUITE_DHE_RSA_WITH_AES_256_CBC_SHA
- CIPHERSUITE_RSA_WITH_AES_128_CBC_SHA256
- CIPHERSUITE_RSA_WITH_AES_256_CBC_SHA256
- CIPHERSUITE_DHE_RSA_WITH_AES_128_CBC_SHA256
- CIPHERSUITE_DHE_RSA_WITH_AES_256_CBC_SHA256

The cipher suite is the default cipher suite.

IKEv2/IPsec

The BR-MLX-10Gx4-IPSEC-M interface module supports creation of virtual private network (VPN) using the IPsec protocol. The IKEv2 protocol is used to negotiate the IPsec service parameters for the VPN.

Note
VLL with RSVP is not supported over IPsec tunnel in any mode of operation in the device.

Note
For Extreme MLXe series devices, the operator shall always enter a minimum 112-bit IKEv2 Pre-Shared Key (PSK).

IPsec critical security parameters

The following parameters make up the IPsec critical security parameters.

- IKEv2 DH Group-14 Private Key 2048 bit MODP
- IKEv2 DH Group-14 Shared Secret 2048 bit MODP

- IKEv2 DH Group-14 Public Key 2048 bit MODP
- IKEv2 ECDH Group-19 Private Key (P-256)
- IKEv2 ECDH Group-19 Shared Secret (P-256)
- IKEv2 ECDH Group-19 Public Key (P-256)
- IKEv2 ECDH Group-20 Private Key (P-384)
- IKEv2 ECDH Group-20 Shared Secret (P-384)
- IKEv2 ECDH Group-20 Public Key (P-384)
- IKEv2 ECDSA Private Key (P-256)
- IKEv2 ECDSA Private Key (P-384)
- IKEv2 ECDSA Public Key (P-256)
- IKEv2 ECDSA Public Key (P-384)
- IKEv2 Encrypt/Decrypt Key
- IKEv2/IPSec Integrity Key
- IKEv2 KDF State
- IKEv2 Pre-Shared Key (PSK)
- IPsec ESP Encrypt/Decrypt Key

IS-IS

IS-IS allows peer-to-peer authentication or client-to-server authentication.

To authorize an authentication, use commands such as the following to configure shared secret keys for IS-IS:

```
device(config)# auth-mode md5 level-1
```

```
device(config)# auth-key jdoepass level-1
```

For more information on IS-IS authentication commands, refer to the *Extreme NetIron Routing Configuration Guide*.

L2 over IPsec

Layer 2 over IPsec supports encryption and decryption of layer 2 (VLL) traffic transmitted or received from the external networks.

For more information on L2 Over IPsec related commands, refer to the Extreme NetIron Routing Configuration Guide.

MACsec

The MACsec protocol is used for securing communication among the trusted components of a 802.1 LAN.

MACsec standards consists of two main components:

- MAC security (MACsec)
- MACsec Key Agreement (MKA) protocol

The MKA protocol defined as part of IEEE 802.1x-2010 standard is responsible for generating the Secure Association Keys (SAK) used by MACsec for symmetric cryptography. This protocol runs on the management card in the control plane.

When MACsec is used to secure the communication between endpoints on a LAN, each packet on the wire is encrypted in the PHY in the data plane using symmetric key cryptography so that communication cannot be monitored or altered on the wire.

MACsec critical security parameters

The following parameters make up the MACsec critical security parameters (CSPs):

- MKA Connectivity Association Key (CAK): Either configured manually by the user or derived from the MSK obtained from the authentication server.
- MKA Connectivity Key Name (CKN): Either configured manually by the user or derived from the EAP session ID obtained from the authentication server.
- MKA Secure Association Key (SAK): Derived from the CAK and used for encryption and decryption of the traffic.
- MKA Integrity Checksum Key (ICK): Derived from SP800-108 KDF.
- MKA Key Encryption Key (KEK): Derived from SP800-108 KDF.
- MKA SP800-108 KDF State

MPLS

Multiprotocol Label Switching (MPLS) allows peer-to-peer authentication or client-to-server authentication. To authorize an authentication, use commands such as the following to configure shared secret keys for MPLS.

For MPLS RSVP:

```
device(config)# rsvp-authentication key jdoepass
```

For MPLS LDP:

```
device(config)# session 10.10.10.3 key jdoepass
```

For more information on MPLS authentication commands, refer to the *Extreme* NetIron *Routing Configuration Guide*.

NTP

Extreme NetIron FIPS devices support Network Time Protocol (NTP) using SHA1.

```
device (config-ntp)# authentication-key key-id 1 sha1
```

Syntax: **[no] authentication-key key-id** *decimal* **sha1**

The following parameter is the NTP critical security parameter (CSP):

- NTP secret

Note
FIPS mode and CC mode do not support MD5 hash algorithm.

OpenFlow

OpenFlow is supported in the FIPS mode as well as the non-FIPS mode.

Note that there is a limit of 3 controllers that can be configured.

The following configuration sets up openflow controller with the FIPS enable system:

```
scp sc-privkey.pem <crypto-officer>@<device-ip-address>:sslclientprivkey
scp sc-cert.pem <crypto-officer>@<device-ip-address>:sslclientcert

device# configure terminal
device(config)# openflow enable ofv130
Warning: Please configure [system-max openflow-flow-entries #] to accept any flows
Warning: Please configure [system-max openflow-pvlan-entries #] to accept Protected VLANs
for Hybrid ports
Warning: Please configure [system-max openflow-unprotectedvlan-entries #] to accept
Configured Unprotected VLANs for Hybrid ports
device(config)# openflow controller ip-address 10.20.180.87  port 600


STEP 2:Now run the below command on the controller(10.20.180.87) -- root/pass
-----------------------------------------------------------------------
[root@centos-180-87 ~]# ./openvswitch-2.3.0/tests/test-controller pssl:600 -p /usr/
local/var/lib/openvswitch/pki/controllerca/ctl-privkey.pem -c /usr/local/var/lib/
openvswitch/pki/controllerca/ctl-cert.pem -C /usr/local/var/lib/openvswitch/pki/switchca/
cacert.pem -O OpenFlow13  -v^C
[root@centos-180-87 ~]# pwd
/root
[root@centos-180-87 ~]#

STEP 3:Observe the below console message and the show command output
-----------------------------------------------------------------------
device(config)# logging console
SYSLOG: <13>Jul 26 19:32:13 OpenFlow: Established active connection with controller
10.20.180.87 port 600.

device(config)#
device# show openflow controller
Openflow controller information

-------------------------------------------------------------------------------
  Controller   Mode      TCP/SSL   IP-address        Port   Status
-------------------------------------------------------------------------------
  1  (Equal)   active    SSL       10.20.180.87      600    OPENFLOW_ESTABLISHED

device# show ip ssl
Session  Protocol Source IP                              Source Port  Remote
IP                              Remote Port
0        TLS_1_2  10.20.81.103                           633
10.20.180.87                          600
```

```
STEP 4: Perform wireshark captures on the management interfaces for different tls
versions for different ciphers
```

HTTPS-based File Copy

The **copy https** command is supported in the Extreme NetIron 6.0.00aa release. The syntax of the command is: `copy https flash <https-server-ip.address> <remote-port(443)> <remote-file-location> <local-file-location>`

The command is supported on different ciphers for different TLS versions as listed below.

- TLS 1.1 version
  - CIPHERSUITE_RSA_WITH_AES_128_CBC_SHA
  - CIPHERSUITE_RSA_WITH_AES_256_CBC_SHA
  - CIPHERSUITE_DHE_RSA_WITH_AES_128_CBC_SHA
  - CIPHERSUITE_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS 1.2 version
  - CIPHERSUITE_RSA_WITH_AES_128_CBC_SHA
  - CIPHERSUITE_RSA_WITH_AES_256_CBC_SHA
  - CIPHERSUITE_DHE_RSA_WITH_AES_128_CBC_SHA
  - CIPHERSUITE_DHE_RSA_WITH_AES_256_CBC_SHA
  - CIPHERSUITE_RSA_WITH_AES_128_CBC_SHA256
  - CIPHERSUITE_RSA_WITH_AES_256_CBC_SHA256
  - CIPHERSUITE_DHE_RSA_WITH_AES_128_CBC_SHA256
  - CIPHERSUITE_DHE_RSA_WITH_AES_256_CBC_SHA256

OSPFv2

The OSPFv2 protocol uses SHA1 for authentication.

OSPF allows peer-to-peer authentication or client-to-server authentication. To authorize an authentication, use commands such as the following to configure shared secret keys for OSPFv2:

```
device(config-if-e1000-1/1)# ip ospf authentication-key P@$$w0rd
device(config-if-e1000-1/2)# ip ospf md5-authentication key-id 1 key P@$$w0rd
device(config-ospf-router)# area 2 virtual-link 2.3.4.5 md5-authentication key-id 2 key P@
$$w0rd
device(config)# ipv6 ospf authentication ipsec spi %u esp sha1 encrypt #on-o
```

For more information on OSPFv2 authentication commands, refer to the *Extreme NetIron Routing Configuration Guide*.

OSPFv3

OSPFv3 supports SHA1 authentication.

To authorize an authentication, use commands such as the following to configure shared secret keys for OSPFv3:

```
device(config)#ipv6 ospf authentication ipsec spi 400 esp sha1
1234567890abcde12345098765432112345678 90
```

For more information on OSPFv3 authentication commands, refer to the *Extreme NetIron Routing Configuration Guide*.

PKI

Public Key Infrastructure (PKI) operates on the management module that allows automated certificate authentication during the IKEv2 session setup. IKEv2 sessions are established on the BR-MLX 10Gx4 IPSEC M interface module.

The following parameters make up the PKI critical security parameters (CSPs):

- PKI SCEP Enrollment RSA 2048-bit Private Key
- PKI SCEP Enrollment RSA 2048-bit Public Key

Note
Extreme NetIron device supports PKI Offline enrollment starting NetIron 6.0.00aa release.

For more information about PKI and IKEv2, refer to the specific sections in the *Extreme NetIron Security Configuration Guide*.

Proprietary 2-way encryption algorithms

The routing protocols OSPFv2, IS-IS, BGP, MPLS LDP, and MPLS RSVP, and the management protocol SNMP save authentication parameters using one of the following two proprietary algorithms:

- Global encoding scheme
- Base 64 encoding scheme

These proprietary algorithms are not supported in FIPS mode but are considered as plain text. When the default FIPS policy is applied, these authentication parameters are zeroized.

RADIUS

HMAC-MD5 authentication used in RADIUS is allowed in FIPS mode.

RADIUS allows peer-to-peer authentication or client-to-server authentication.

Radius over TLS is supported in the FIPS mode.

The following parameter makes up the RADIUS critical security parameter (CSP):

- RADIUS Secret

Note

For more information on RADIUS authentication commands, refer to the *Extreme NetIron Command Reference* and the *Extreme NetIron Routing Configuration Guide*.

Application timer

When TLS is used with OCSP during chain certificate validation or when stunnel is used as proxy TLS server for RADIUS, it is recommended to maximize the connection timeout for RADIUS. RADIUS timeout can be set to a maximum value of 12 seconds using the following command.

```
config# radius-server-timeout <val 3-12secs>
```

SCP

The following table lists the Secure Copy (SCP) commands that are available to compensate for equivalent existing functionality of TFTP commands disabled in FIPS mode.

Table 7: Corresponding TFTP and SCP commands

Command functionality	TFTP commands not allowed in FIPS mode	SCP commands with corresponding functionality in FIPS mode
Import a digital certificate	**ip ssl certificate-data-file tftp** *ip-address certificate-filename*	**scp** *certificate-filename* user@*ip-address*:sslCert
Import an RSA private key from a client	**ip ssl private-key-file tftp** *ip-address key-filename*	**scp** *key-filename* user@*ip-address*: sslPrivKey
Load an RSA public key file from a client	**ip ssh pub-key-file tftp** *ip-address key-filename*	**scp** *key-filename* user@ *ip-address*: sshPubKey

Importing a digital certificate

To import a digital certificate using SCP, enter a command such as the following:

```
C:> scp certfile user@192.168.89.210:sslCert
```

Syntax: **scp** *certificate-filename* user@*ip-address*:sslCert

Note
The **scp** command is not supported on NetIron CER devices.

The *certificate-filename* variable is the file name of the digital certificate that you are importing to the device.

The *ip-address* variable is the IP address of the server from which the digital certificate file is downloaded.

The functionality of the **scp** command is equivalent to that of the **disabled ip ssl certificate-data-file tftp** command.

For more information on the **scp** command, refer to the *Extreme NetIron Routing Configuration Guide*.

Importing an RSA private key from a client

To import an RSA private key from a client using SCP, enter a command such as the following:

```
C:> scp keyfile user@192.168.9.210:sslPrivKey
```

Syntax: **scp** *key-filename* user@*ip-address*:sslPrivKey

Note
The **scp** command is not supported on NetIron CER devices.

The *key-filename* variable is the file name of the private key that you want to import into the device.

The *ip-address* variable is the IP address of the server that contains the private key file.

The functionality of the **scp** command is equivalent to that of the **disabled ip ssl private-key-file tftp** command.

For more information on the **scp** command, refer to the *Extreme NetIron Routing Configuration Guide*.

SNMP

In the FIPS mode of operation, the device uses the existing SNMP configuration. However, MIB objects related to keys and passwords output NULL or a 0 value.

Note
SNMPv1 and SNMPv2C versions are not allowed in FIPS mode. Access is allowed only for SNMPv3 configuration with authPriv mode. Other security modes such as noAuthNoPriv and authNoPriv are not allowed.

SNMP allows peer-to-peer authentication or client-to-server authentication. To authorize an authentication, use commands such as the following to configure shared secret keys for SNMP:

```
device(config)# snmp-server community extremeSNMP
```

SNMP notification

In the FIPS mode or CC mode of operation, the Extreme NetIron device generates only SNMPv3 notifications if it has to be configured for SNMPv3 host in authPriv security mode. As a result, both authentication and privacy are configured for a given SNMP target.

Note
The device does not validate any configuration of **snmp-server host** command to ensure SNMPv3 authPriv configuration. During the notification generation instance, the system goes through the configured SNMP host list and sends notification to only those hosts that have SNMPv3 with authPriv security mode.

SNMP CSP objects

The following SNMP MIB objects represent the critical security parameter (CSP) entities that are restricted in FIPS mode.

Enterprise MIB objects:

- snRadiusKey
- snRadiusServerRowKey
- snTacacsKey
- snTacacsServerRowKey
- snVrrpIfAuthPassword
- snAgGblPassword
- snAgGblReadOnlyCommunity
- snAgGblReadWriteCommunity
- snAgGblTelnetPassword
- snAgentUserAccntPassword

Standard MIB objects:

- rip2IfConfAuthKey
- vrrpOperAuthKey
- dvmrpInterfaceKey
- ospfIfAuthKey
- ospfVirtIfAuthKey

SSHv2

Secure Shell version 2 (SSHv2) is allowed in FIPS mode.

The following SSH commands are affected when the Extreme device is in FIPS mode:

- The **ssh server** command enables the SSH server. The SSH server is always enabled; however, to start it, use the **crypto key generate** command to create host keys.
- The **ip ssh encryption aes-only** command is disabled.

  During SSH connection, encryption is done using AES 256 or AES 128, depending on client's capability.
- The **ip ssh key-authentication** command is disabled.
- The **ip ssh permit-empty-password** command is disabled.
- The **ip ssh pub-key-file tftp** command is disabled.
- The **ip ssh scp** command ensures that SCP is enabled to run in FIPS mode. SCP is needed for file communication and the **ip ssh scp disable** command is disabled in FIPS mode and displays the following message:

  ```
  FIPS Compliance: SCP needs to be enabled
  ```
- The **crypto key zeroize** command removes configured SSH keys.

Use the **show ip ssh config** command to display SSH configuration information.

For more information on the **show ip ssh config** command, refer to the *Extreme NetIron Security Configuration Guide*.

SSH key generation time is affected by the increased security of authentication and encryption algorithms both in and out of FIPS mode.

The **ip ssh password-authentication** [**no** | **yes**] command is used to disable the password authentication for SSH. The **ip ssh interactive-authentication** [**no** | **yes**] command is used to disable the interactive authentication for SSH. For more information about these commands, refer to the *Extreme NetIron Security Configuration Guide*.

The following table shows the supported SSH ciphers.

Table 8: SSH ciphers supported by NetIron devices

Extreme NetIron release	SSH cipher supported
Pre-5.8 FIPS mode	aes256-cbc and aes128-cbc
5.8 and later FIPS mode	aes256-ctr, aes192-ctr, aes128-ctr, aes256-cbc,aes192-cbc, and aes128-cbc
5.8 and later JITC mode	aes256-ctr, aes192-ctr, and aes128-ctr
5.8 and later CC mode	aes256-cbc and aes128-cbc

The following parameters make up the SSHv2 critical security parameters (CSPs):

- SSHv2 Client RSA Private Key
- SSHv2 Client RSA Public Key
- SSHv2 DH Group-14 Peer Public Key 2048 bit MODP
- SSHv2 DH Group-14 Private Key 2048 bit MODP
- SSHv2 DH Group-14 Public Key 2048 bit MODP
- SSHv2 DH Shared Secret Key (2048 bit)
- SSHv2 Host RSA Private Key (2048 bit)
- SSHv2 Host RSA Public Key (2048 bit)
- SSHv2 KDF Internal State
- SSHv2/SCP Authentication Key (HMAC-SHA-1, 160 bits)
- SSHv2/SCP Session Keys (128, 192 and 256 bit (AES CBC and AES CTR)

Refer to the Extreme NetIron configuration guides for SSH key generation time ranges.

Syslog

Syslog is a standard service for message logging. It permits separation of the software that generates messages, the system that stores them, and the software that reports and analyzes them. This is an implicit service configured by the Crypto-officer role. This service can be used to view the syslog audit records saved on the cryptographic module.

TACACS+

TACACS+ allows peer-to-peer authentication or client-to-server authentication.

MD5-based operator authentication used in TACACS+ is allowed in FIPS mode. To authorize an authentication, use commands such as the following to configure shared secret keys for TACACS+:

```
device(config)# tacacs-server key <string>
```

The following parameter makes up the TACACS+ critical security parameters (CSP):

- TACACS+ Secret

For more information on TACACS+ authentication commands, refer to the Extreme NetIron *Routing Configuration Guide*.

Telnet

Telnet is disabled in FIPS mode as part of the default FIPS policy on the device. Attempts to start the Telnet server fail in FIPS mode.

TFTP

The following TFTP commands are disabled and return an error when TFTP operation is not allowed on the device in FIPS mode:

- All **copy tftp** commands
- The **boot system tftp** *ip-address filename* command
- The **boot system auxiliary flash** *file* command

The following TFTP commands are disabled. Use SCP commands with equivalent functionality instead. Refer to SCP.

- **ip ssl certificate-data-file tftp** *ip-address certificate-filename*
- **ip ssl private-key-file tftp** *ip-address key-filename*
- **ip ssh pub-key-file tftp** *ip-address key-filename*

VRRP

Virtual Router Redundancy Protocol (VRRP) is an election protocol that provides redundancy to routers within a LAN. VRRP allows you to provide alternate router paths for a host without changing the IP address or MAC address by which the host knows its gateway.

VRRP in Layer 3

Execution of this service in the Layer 3 mode (plaintext) is supported only in the non-approved mode. Extreme devices support plain text authentication.

VRRP-E

Virtual Router Redundancy Protocol Enhanced (VRRP-E) is the proprietary version of VRRP that overcomes limitations in the standard protocol.

VRRP-E in Layer 3

Execution of this service in the Layer 3 mode (plaintext) is supported only in the non-approved mode. Extreme devices support plaintext and HMAC MD-5 authentication.

Web Management

Web Management is not supported when FIPS mode is enabled on the device.

DRBG Health Test on IPsec MP

Deterministic Random Bit Generator (DRBG) health and error checks are performed on the management module (MP) crypto module used in the Brocade NetIron MLXe device

The FIPS self-test is executed at system startup, which includes DRBG health and error checks. This startup test executes a known answer test, which includes DRBG health and error checks.

DRBG tests are performed on demand by the user by using the following command:

```
fips crypto drbg
```

The expected result is the test is passed. In the event of failure, the system will restart, and perform the test again as part of FIPS self-tests executed at system startup.

The DRBG Known Answer Test (KAT) and health test are performed during:

- System boot-up and at regular intervals.
- On-demand and periodic testing after 2^24 uses, during instantiate and reseed.
- DRBG check immediately after powering on the system.

The type of DRBG mechanism and the cryptographic primitives used (for example: AES-128 or SHA-256), are as follows:

- Type of DRBG mechanism: CTR_DRBG
- Cryptographic primitives used: AES-256

Security strengths of the cryptographic algorithms supported by the implementation: AES-256

The features (such as prediction resistance, personalization string, additional input) supported by this implementation are as follows:

- Prediction Resistance is TRUE.
- Personalization String
- Additional Input

Note

The DRBG mechanism functions are not distributed. In the case of `CTR_DRBG`, a derivation function is used. The code used to perform the DRBG Health Test on IPSec MP is from OpenSSL FIPS205.

Command example

DRBG functions can be tested on a demand basis as shown in the following CLI example.

```
device# fips crypto drbg
generating 1024 random bytes
generating random signed 32 bit and 64 bit values
random 32 bit int 1961644244 and 64 bit 1870544140
generating random unsigned 32 bit and 64 bit values
random unsigned int 338113164 and 64 bit 63160224
cli_fips_crypto_drbg successful.
device# fips crypto force-failure drbg
Random number generation success and setting force failure
device# fips crypto drbg

Failed to generate 1024 random bytes

SYSLOG: <10>Sep  6 19:17:17 FIPS Fatal Cryptographic Module Failure. Reason: Generic
NetIron XMR/MLX Boot Code Version 5.9.0
..MPP.
Enter 'a' to stop at memory test
Enter 'b' to stop at boot monitor
```

System reset and boot up in FIPS mode

POST testing takes place as the device progresses through the boot sequence.

The following actions and limitations take effect when the device is operationally in FIPS mode according to the FIPS default policy:

- Boot up from TFTP is disabled.
- The monitor mode memory access command set is disabled. Configure an alternative FIPS policy to the default policy to access the command set. Refer to Modifying the FIPS policy.
- Boot monitor access during a cold boot is disabled with the exception of the option to access monitor mode during the boot sequence. Refer to Accessing monitor mode in the event of continuous failure.
- Access to memory test mode is disabled.
- Debug commands are disabled from the application prompt in FIPS mode.

Debugging in FIPS mode

The device reloads automatically when it encounters a system reset and enters FIPS failure state. The cause of failure logs on the console and the device performs a self-reboot.

You can conduct debugging in monitor mode when a flexible FIPS policy is applied on the device and in the event of continuous failure. Refer to Access to monitor mode.

Placing the device in FIPS mode

Placing the device in FIPS mode is a multiple-step process that begins with enabling FIPS mode on the device.

This places the device administratively in FIPS mode. To operate the device in FIPS mode, save the configuration, and reboot the device. Always back up the desired configuration to ensure it is saved in the event of a system reset.

General steps to place the Extreme NetIron device in FIPS mode

Perform the following steps to place the Extreme NetIron device in FIPS mode.

Procedure

1. Assume the Crypto-officer role.
2. Copy the needed signature files. Refer to Copying the signature files.
3. Enable FIPS mode. Refer to Enabling FIPS mode. The device enables FIPS administrative commands. The device is not in the FIPS approved mode yet. Do not change the default strict FIPS security policy, which is required for the FIPS approved mode.
4. Zeroize shared secrets and host keys. Refer to Zeroizing shared secrets and host keys.
5. Configure all users of the module and the authentication methods. Refer to Configuring user authentication
6. Save the configuration. Refer to Saving the configuration.
7. Reload the device. Refer to Reloading the device.
8. Enter the **`fips show`** command. The device displays the FIPS-related status, which should confirm the security policy is the default security policy.
9. Perform a FIPS self-test to verify the correct signature files were copied. Refer to Perform a FIPS self-test.
10. Inspect the physical security of the module including placement of tamper evident labels on the Extreme NetIron device. Refer to the *Extreme FIPS Security Seal* document for more information.

Copying the signature files

As part of placing the device in FIPS mode, you should copy the specific signature files into the device.

Refer to the *Extreme NetIron Software Upgrade Guide* for the required signature file information.

When the NetIron device is in FIPS mode, the RSA2048-SHA256-based signature firmware integrity check is done during the image installation and during image reload. For the firmware integrity check and the device reload to be successful, always retain the signature files that were copied to the device at image installation time.

Note

The device may not reload if you do not retain the signature files or if you copy invalid signature files.

For the MLX Series devices, the signature files in the following table must be loaded to the management module with specific destination file names.

Note
Where the .sig extension appears in the source file name, you can use either .sig or .sha256. Use .sig if the device is running NetIron 5.6.00a or earlier. Use .sha256 if the device is running NetIron 5.6.00aa or later.

Table 9: Required signature files for the Extreme MLXe devices

Image name on flash	Image type	Signature source file name	Signature destination file name	RSA2048/ SHA256 bit signature source file name
primary	Management Application	xmrXXXXX.sig	primary.sig	xmrXXXXX.sha256
secondary	Management Application	xmrXXXXX.sig	secondary.sig	
Monitor	Management Monitor	xmbXXXXX.sig	monitor.sig	xmbXXXXX.sha256
lp-monitor	Interface Module Monitor	xmlbXXXXX.sig	lp-mon.sig	xmlbXXXXX.sha256
p-primary-0	Interface Module Application	xmlpXXXXX.sig	lp-pri.sig	xmlpXXXXX..sha256
lp-secondary-0	Interface Module Application	xmlpXXXXX.sig	lp-sec.sig	

For the NetIron CER devices, the signature files in the following table must be loaded to the management module with specific destination file names.

Table 10: Required signature files for the NetIron CER devices

Image name on flash	Image type	Signature source file name	Signature destination file name	RSA2048/ SHA256 bit signature source file name
primary	Management Application	ceXXXXX.sig	primary.sig	ceXXXXX.sha256
secondary	Management Application	ceXXXXX.sig	secondary.sig	ceXXXXX.sha256
Monitor	Management Monitor	cebXXXXX.sig	monitor.sig	cebXXXXX.sha256

Note
The signature files are specific to the version of the images currently in the flash code of the device.

Note

The **fips policy allow tftp-access** command must be enabled if FIPS is enabled using the TFTP commands.

Copying signature files for ExtremeCER 2000-4X devices

Procedure

1. Place the needed signature files on an accessible SCP or TFTP server.
2. Copy the management monitor image signature file by entering one of the following commands:
   - Using SCP on a remote client:

     scp ceb*xxxxx***.sig** *user@device-IpAddress*:**flash:monitor.sig**
   - Using TFTP at the Privileged EXEC level of the CLI:

     copy tftp flash *tftp-srvr***ceb** *xxxxx***.sig monitor.sig**
3. Copy the management module application image signature file by entering one of the following commands:
   - Using SCP on a remote client:

     scp ce*xxxxx***.sig** *user@device-IpAddress*:**flash:**[*primary.sig* | *secondary.sig*]
   - Using TFTP at the Privileged EXEC level of the CLI:

     copy tftp flash *tftp-srvr* **ce***xxxxx***.sig** [*primary.sig* | *secondary.sig*]
4. Copy the application image file by entering one of the following commands:
   - Using SCP on a remote client:

     scp ce*xxxxx***.sig** *user@device-IpAddress*:**flash:primary**
   - Using TFTP at the Privileged EXEC level of the CLI:

     copy tftp flash *tftp-srvr* **ce***xxxxx***.bin** [*primary* | *secondary*]

Copying the signature files for Extreme NetIron MLXe devices

Procedure

1. Place the needed signature files on an accessible SCP or TFTP server.
2. Copy the management monitor image signature file by entering one of the following commands:
   - Using SCP on a remote client:

     scp xmb*xxxxx* **.sig***user@device-IpAddress*:**flash:monitor.sig**
   - Using TFTP at the Privileged EXEC level of the CLI:

     copy tftp flash *tftp-srvr* **xmb***xxxxx***.sig monitor.sig**

3. Copy the interface module monitor image signature file by entering one of the following commands:
   - Using SCP on a remote client:

     scp xmlb*xxxxx***.sig** *user@device-IpAddress*:**flash:lp-mon.sig**
   - Using TFTP at the Privileged EXEC level of the CLI:

     copy tftp flash *tftp-srvr* **xmlb***xxxxx***.sig lp-mon.sig**
4. Copy the interface module application image signature file by entering one of the following commands:
   - Using SCP on a remote client:

     scp xmlp*xxxxx***.sig** *user@device-IpAddress*:**flash:**[*lp-pri.sig* | *lp-sec.sig*]
   - Using TFTP at the Privileged EXEC level of the CLI:

     copy tftp flash *tftp-srvr* **xmlp***xxxxx***.sig** [*lp-pri.sig* | *lp-sec.sig*]
5. Copy the management module application image signature file by entering one of the following commands:
   - Using SCP on a remote client:

     scp xmr*xxxxx***.sig** *user@device-IpAddress*:**flash:**[*primary.sig* | *secondary.sig*]
   - Using TFTP at the Privileged EXEC level of the CLI:

     copy tftp flash *tftp-srvr* **xmr***xxxxx***.sig** [*primary.sig* | *secondary.sig*]

Enabling FIPS mode

Perform the following steps to enable FIPS mode.

Procedure

1. Attach a management station (PC or terminal) to the management module serial (console) port using a serial cable.

   When the device is not in a console session, FIPS-related commands return errors.
2. Verify that the device is in non-FIPS mode by using the **fips show** command.

   ```
   device(config)# fips show
   ```

   The **fips show** command lists the current configuration of the device and can be run in both FIPS mode and non-FIPS mode to establish whether the device is truly in FIPS mode.

   The output of the **fips show** command confirms that the device is in FIPS mode and identifies the device as either administratively or operationally in FIPS mode.

   Note

   If the Extreme device is in JITC mode, then you cannot enable FIPS on the device.

   The following example shows the output of the **fips show** command before the **fips enable** command is entered, and administrative status is off and operational status is off:

For MLX device:

```
device# fips show
FIPS Validated Cryptographic Module
MP FIPS Version: EXTR-NI-IP-CRYPTO-VER-4.0
LP FIPS Version: EXTR-NI-LP-CRYPTO-VER-2.0
LP IPsec FPGA FIPS Version: EXTR-NI-LP-FPGA-CRYPTO-VER-1.0
```

For Extreme NetIron CER device:

```
device# fips show
FIPS Validated Cryptographic Module
MP FIPS Version: EXTR-NI-IP-CRYPTO-VER-4.0
LP FIPS Version: EXTR-NI-LP-CRYPTO-VER-2.0
```

If the device is already in administrative FIPS mode, you can modify the FIPS policy. Refer to Modifying the FIPS policy.

3. Use the **fips enable** command to place the device administratively in FIPS mode.

```
device(config)# fips enable
WARNING: This will enable FIPS on this device. Please refer
: to the NetIron Federal Information Processing Standards Guide for
: more details. Also, be advised that Software/Firmware Integrity checks
: will always be performed on this device on subsequent reloads, even
: if FIPS or Common Criteria is disabled in the future.
Are you sure? (enter 'y' or 'n'): y
device(config)# fips enable
```

Syntax: **[no] fips enable**

The following example shows the output of the **fips enable** command on MLX Series devices.

```
device(config)# fips enable
WARNING: This will enable FIPS on this device. Please refer
       : to the NetIron Federal Information Processing Standards Guide for
       : more details. Also, be advised that Software/Firmware Integrity checks
       : will always be performed on this device on subsequent reloads, even
       : if FIPS or Common Criteria is disabled in the future.
Are you sure? (enter 'y' or 'n'): y
This device is now running in FIPS administrative mode.
At this time you can alter this system's FIPS default security policy
and then enter FIPS operational mode.

Note: Making changes to the default FIPS security policy weakens
the security of the device and makes the device non-compliant with
FIPS 140-2 Level 2, design assurance Level 3
The default security policy defined in the FIPS
Security Policy Document ensures that the device complies with all
FIPS 140-2 specifications. Commands to alter the default security policy
are available to the crypto-officer; however, <ph varref="Company_Name">Extreme</ph>
does not recommend
making changes to the default security policy at any time.
====================================

To enter FIPS mode, complete the following steps:
1. Optionally, configure FIPS policy commands that meets your network
   requirements. You must explicitly configure the following services if you
   want to use them when the device is operational in FIPS mode:
      - Allow TFTP access.
          Current status: Enabled
      - Allow SNMP Access to the Critical Security Parameter (CSP) MIB objects.
          Current status: Disabled
```

```
      - Allow access to all commands within the monitor mode.
          Current status: Disabled
      - Allow cleartext password display in some commands.
          Current status: Disabled
      - Retention of shared secret keys for all protocols and the host passwords.
          Current status: Retain
      - Retention of SSH RSA host keys.
          Current status: Clear
      - Retention of HTTPS RSA host keys and certificate.
          Current status: Clear

2. Enter the "fips zeroize all" command, which zeroes out the shared secrets
    used by various networking protocols, including the host access passwords,
    SSH and HTTPS host-keys with the digital signature based on the configured
    FIPS Security Policy.
3. Save the running configuration.
4. Reload the device.
5. Enter the "fips show" command to verify that the device entered
  FIPS or CC operational mode.
====================================

In FIPS mode, the system will disable the following services or commands after
reload:
FIPS. Telnet server will be disabled.
      The "telnet server" command will be removed.
FIPS. SSL Client will be enabled.
FIPS. TLS version 1.0 will be disabled.(Applicable for MLX & CER
devices)
FIPS. SCP will be enabled.
      The "ip ssh scp disable" command will be removed.
FIPS. FIPS Configuration "boot system {slot1|slot2} <file>" will be removed
      as FIPS mode does not allow system to boot from Storage Card.
FIPS. Configuration "lp boot system {slot1|slot2} <file> <slot>" will be
      removed as FIPS mode does not allow system to boot from Storage Card.
FIPS. Configuration "boot system tftp <ip> <file>" will be removed as FIPS
      mode does not allow system to boot from TFTP.
FIPS. Configuration "enable password-display" will be removed.
FIPS. HTTP server will be disabled. The "web-management http" command will
      be removed.
FIPS. HTTPS server will change as follows:
      -SSL 3.0 and TLS 1.0 will be disabled.
      -TLS version 1.1 and greater will be used.(Applicable only for MLX devices)
      -RC4 cipher will be disabled.
      -Passwords will be required; the "web-management allow-no-password"
        command will be removed.
FIPS. SNMP server will change as follows:
      -SNMP support for v1 and v2 versions will be disabled.
      -For SNMPv3 version authentication and privacy is mandatory,
       and MD5 authentication key and DES privacy password will be disabled.
FIPS. NTP md5 authentication will be disabled.
FIPS. HTTP Client will be disabled.
FIPS. Passwords/Keys which don't comply with FIPS standards will be removed
      on reload.
FIPS. Please see FIPS config guide for complete details.

FIPS. Configuration "enable aaa console" will be disabled temporarily to
      allow console access to configure SSH parameters. It can be
      re-enabled after SSH is confirmed operational
      Current status of "enable aaa console" is: Disabled

====================================
Additionally, in FIPS  only operational mode, the system will have the
following restrictions
FIPS. Configuration for CLI logging "logging cli-command" will be removed.
```

The following example shows the output of the **fips enable** command on the Extreme NetIron CER devices.

```
device(config)# fips enable
WARNING: This will enable FIPS on this device. Please refer
       : to the NetIron Federal Information Processing Standards Guide for
       : more details. Also, be advised that Software/Firmware Integrity checks
       : will always be performed on this device on subsequent reloads, even
       : if FIPS or Common Criteria is disabled in the future.
Are you sure? (enter 'y' or 'n'): y
This device is now running in FIPS administrative mode.
At this time you can alter this system's FIPS default security policy
and then enter FIPS operational mode.

Note: Making changes to the default FIPS security policy weakens
the security of the device and makes the device non-compliant with
FIPS 140-2 Level 2, design assurance Level 3
The default security policy defined in the FIPS
Security Policy Document ensures that the device complies with all
FIPS 140-2 specifications. Commands to alter the default security policy
are available to the crypto-officer; however, <ph varref="Company_Name">Extreme</ph>
does not recommend
making changes to the default security policy at any time.
====================================

To enter FIPS mode, complete the following steps:
1. Optionally, configure FIPS policy commands that meets your network
   requirements. You must explicitly configure the following services if you
   want to use them when the device is operational in FIPS mode:
      - Allow TFTP access.
          Current status: Enabled
      - Allow SNMP Access to the Critical Security Parameter (CSP) MIB objects.
          Current status: Enabled
      - Allow access to all commands within the monitor mode.
          Current status: Enabled
      - Allow cleartext password display in some commands.
          Current status: Disabled
      - Retention of shared secret keys for all protocols and the host passwords.
          Current status: Retain
      - Retention of SSH RSA host keys.
          Current status: Retain

2. Enter the "fips zeroize all" command, which zeroes out the shared secrets
    used by various networking protocols, including the host access passwords,
    SSH and HTTPS host-keys with the digital signature based on the configured
    FIPS Security Policy.
3. Save the running configuration.
4. Reload the device.
5. Enter the "fips show" command to verify that the device entered
  FIPS or CC operational mode.
====================================

In FIPS mode, the system will disable the following services or commands after
reload:
FIPS. Telnet server will be disabled.
      The "telnet server" command will be removed.
FIPS. SSL Client will be enabled.
FIPS. SCP will be enabled.
      The "ip ssh scp disable" command will be removed.

FIPS. SNMP server will change as follows:
      -SNMP support for v1 and v2 versions will be disabled.
      -For SNMPv3 version authentication and privacy is mandatory,
       and MD5 authentication key and DES privacy password will be disabled.
```

```
FIPS. NTP md5 authentication will be disabled.
FIPS. HTTP Client will be disabled.
FIPS. Passwords/Keys which don't comply FIPS standards will be removed
      on reload.
FIPS. Please see FIPS config guide for complete details.

FIPS. Configuration "enable aaa console" will be disabled temporarily to
      allow console access to configure SSH parameters. It can be
      re-enabled after SSH is confirmed operational
      Current status of "enable aaa console" is: Disabled

====================================
Additionally, in FIPS  only operational mode, the system will have the
following restrictions
FIPS. Configuration for CLI logging "logging cli-command" will be removed.
device#
```

4. Verify the status of the device as administratively in FIPS mode by using the **fips show** command.

   The following example shows the output of the **fips show** command on a MLX Series device after the **fips enable** command is entered and administrative status is on and operational status is off.

```
device# fips show
FIPS Validated Cryptographic Module
MP FIPS Version: EXTR-NI-IP-CRYPTO-VER-4.0
LP FIPS Version: EXTR-NI-LP-CRYPTO-VER-2.0
FIPS mode   : Administrative status ON: Operational status ON
FIPS CC mode: Administrative status OFF: Operational status OFF

System Specific:
OS monitor access status is: Disabled

Management Protocol Specific:
Telnet server                 : Disabled
Telnet client                 : Disabled
TFTP client                   : Enabled
HTTPS SSL 3.0                 : Disabled
SNMP v1, v2, v2c              : Disabled
SNMP Access to security objects: Disabled
Password Display              : Disabled
Critical security Parameter updates across FIPS boundary:
(i.e. during "fips zeroize" ..., or "no fips enable")   :
Protocol Shared secret and host passwords: Retain
SSH RSA Host keys                       : Clear
HTTPS RSA Host Keys and Signature       : Clear
```

   The following example shows the output of the **fips show** command on an Extreme NetIron CER device after the **fips enable** command is entered and administrative status is on and operational status is off:

```
device# fips show
FIPS Validated Cryptographic Module
MP FIPS Version: EXTR-NI-IP-CRYPTO-VER-4.0
FIPS mode   : Administrative status ON: Operational status ON
FIPS CC mode: Administrative status OFF: Operational status OFF

System Specific:
OS monitor access status is: Disabled

Management Protocol Specific:
Telnet server                 : Disabled
Telnet client                 : Disabled
TFTP client                   : Enabled
SNMP v1, v2, v2c              : Disabled
```

```
SNMP Access to security objects: Disabled
Password Display               : Disabled
Critical security Parameter updates across FIPS boundary:
(i.e. during "fips zeroize" ..., or "no fips enable")   :
Protocol Shared secret and host passwords: Clear
SSH RSA Host keys                        : Clear
```

Zeroizing shared secrets and host keys

After you have reviewed the FIPS policy, use the **fips zeroize all** command to zeroize all plain text secrets, private keys and CSPs.

```
device# fips zeroize all
```

Syntax: **[no] fips zeroize** {**all** | **shared-secret** | **host-keys**}

The **all** option zeroizes all shared secrets and host keys. The **shared-secret** option zeroizes shared secret keys only. The **host-keys** option zeroizes host keys only.

It displays KCM key string deletion.

For example,**key-string for key-id <key-id>** under **key chain <keychain-name>** is deleted successfully.

For example, entering **fips zeroize shared-secret** command zeroizes only the shared secret keys of various networking protocols and host access passwords.

Note
The **fips zeroize all** command may cause operational failure within networking protocols using shared secrets and should be used with careful consideration.

The default FIPS policy calls for the zeroization of all keys using the **fips zeroize all** command option. When you apply a less strict FIPS policy than the default, zeroize at your discretion.

Note
If there are any errors displayed during zeroization for pending SSH or HTTPS sessions in use, then let the Crypto-officer clear the corresponding sessions by either disconnecting the clients remotely or by using the **kill ssh** or **clear web-connection** commands.

Note
Run the **clear ikev2 sa** command to manually remove the connection once the FIPS mode is disabled.

Note
The **fips zeroize all** command zeroizes all keys irrespective of the configured FIPS policy.

The following table lists the various keys used in the system that are zeroized in compliance with FIPS.

Table 11: Key zeroization

Keys used
IKEv2 DH Group-14 Private Key 2048 bit MODP
IKEv2 DH Group-14 Public Key 2048 bit MODP
IKEv2 DH Group-14 Shared Secret 2048 bit MODP
IKEv2 ECDH Group-19 Private Key (P-256)
IKEv2 ECDH Group-19 Public Key (P-256)
IKEv2 ECDH Group-19 Shared Secret (P-256)
IKEv2 ECDH Group-20 Private Key (P-384)
IKEv2 ECDH Group-20 Public Key (P-384)
IKEv2 ECDH Group-20 Shared Secret (P-384)
IKEv2 ECDSA Private Key (P-256)
IKEv2 ECDSA Private Key (P-384)
IKEv2 ECDSA Public Key (P-256)
IKEv2 ECDSA Public Key (P-384)
IKEv2 Encrypt/Decrypt Key
IKEv2 KDF State
IKEv2 Pre-Shared Key (PSK)
IKEv2/IPSec Authentication Key
IPsec ESP Encrypt/Decrypt Key
Local - Crypto-officer Password
Local - Port Administrator Password
Local - User Password
LP DRBG Internal State LP DRBG Seed LP DRBG Value C LP DRBG Value V
LP DRBG Internal State
LP DRBG Seed
LP DRBG Value C
LP DRBG Value V
MKA Connectivity Association Key (CAK)
MKA Connectivity Key Name (CKN)
MKA Integrity Checksum Key (ICK)
MKA Key Encryption Key (KEK)
MKA Secure Association Key (SAK)

Table 11: Key zeroization (continued)

Keys used
MKA SP800-108 KDF State
MP DRBG Internal State
MP DRBG Key
MP DRBG Seed
MP DRBG Value V
NTP secret
PKI SCEP Enrollment RSA 2048-bit Private Key
PKI SCEP Enrollment RSA 2048-bit Public Key
RADIUS Secret
SNMPv3 KDF State
SNMPv3 secret
SSHv2 Client RSA Private Key
SSHv2 Client RSA Public Key
SSHv2 DH Group-14 Peer Public Key 2048 bit MODP
SSHv2 DH Group-14 Private Key 2048 bit MODP
SSHv2 DH Group-14 Public Key 2048 bit MODP
SSHv2 DH Shared Secret Key (2048 bit)
SSHv2 Host RSA Private Key (2048 bit)
SSHv2 Host RSA Public Key (2048 bit)
SSHv2 KDF Internal State
SSHv2/SCP Authentication Key (HMAC-SHA-1, 160 bits)
SSHv2/SCP Session Keys (128, 192 and 256 bit (AES CBC and AES CTR)
TACACS+ Secret
TLS Authentication Key
TLS Host DH Group-14 Private Key 2048 bit MODP
TLS Host DH Group-14 Public Key 2048 bit MODP
TLS Host RSA Private Key (RSA 2048 bit)
TLS Host RSA Public Key (RSA 2048 bit)
TLS KDF Internal State
TLS Master Secret
TLS Peer DH Group-14 Public Key 2048 bit MODP
TLS Peer Public Key (RSA 2048 bit)
TLS Pre-Master Secret
TLS Session Key

Configuring user authentication

Extreme NetIron devices support role-based authentication. A device can perform authentication and authorization (role selection) using TACACS+, RADIUS, and local configuration database. NetIron devices also support multiple authentication methods for each service.

To implement one or more authentication methods for securing access to the device, you configure authentication-method lists that set the order in which the authentication methods are consulted.

In an authentication-method list, you specify the access method (SSHv2, Web, SNMP, and so on) and the order in which the device tries one or more of the following authentication methods:

- Line password authentication
- Enable password authentication
- Local user authentication
- RADIUS authentication
- TACACS+ authentication

When a list is configured, the device attempts the first method listed to provide authentication. If that method is not available, (for example, the device cannot reach a TACACS+ server) the device tries the next method until a method in the list is available or all methods have been tried.

NetIron devices allow multiple concurrent operators through SSHv2 and the console. One operator's configuration changes can overwrite the changes of another operator.

Note
The Crypto-officer should enable the password restriction using the **`enable strict-password-enforcement`** command.

Line password authentication

The password authentication method uses the Telnet password to authenticate an operator. To use line authentication, a Crypto-officer must set the Telnet password.

Note
When operating in the FIPS approved mode, Telnet is disabled and line authentication is not available.

Enable password authentication

The enable method uses a password corresponding to each role to authenticate an operator. An operator must enter the read-only password to select the User role. An operator enters the port-config password to the Port Configuration Administrator role. An operator enters the super-user password to select the Crypto-officer Role.

To use enable authentication method, a Crypto-officer must set the password for each privilege level.

Local user authentication

The local method of authentication uses a password associated with a user name to authenticate an operator. An operator enters a user name and corresponding password. The NetIron device assigns the role associated with the user name to the operator when authentication is successful.

To use local authentication, a Crypto-officer must define user accounts. The definition includes a user name, password, and privilege level (which determines the role).

RADIUS authentication

The RADIUS method uses one or more RADIUS servers to verify user names and passwords. The NetIron device prompts an operator for user name and password. The device sends the user name and password to the RADIUS server. Upon successful authentication, the RADIUS server returns the operator's privilege level, which determines the operator's role. If a RADIUS server does not respond, the NetIron device will send the user name and password information to the next configured RADIUS server.

NetIron series devices support additional command authorization with RADIUS authentication. The following events occur when RADIUS command authorization takes place.

1. A user previously authenticated by a RADIUS server enters a command on the NetIron device.
2. The NetIron device looks at its configuration to see if the command is at a privilege level that requires RADIUS command authorization.
3. If the command belongs to a privilege level that requires authorization, the NetIron device looks at the list of commands returned to it when RADIUS server authenticated the user.

After RADIUS authentication takes place, the command list resides on the NetIron device. The device does not consult the RADIUS server again once the operator has been authenticated. This means that any changes made to the operator's command list on the RADIUS server are not reflected until the next time the RADIUS server authenticates the operator, and the server sends a new command list to the NetIron device.

Note

Radius over TLS is supported in the FIPS mode.

To use RADIUS authentication, a Crypto-officer must configure RADIUS server settings along with authentication and authorization settings.

TACACS+ authentication

The TACACS+ methods use one or more TACACS+ servers to verify user names and passwords. For TACACS+, the NetIron device prompts an operator for user name and password. The device sends the user name and password to the TACACS+ server. Upon successful authentication, the NetIron device selects the operator's role implicitly based on the action requested (for example, User role for a login request or Crypto-officer role for a configure terminal command). For TACACS+ authentication, the NetIron device prompts an operator for a user name, which the device uses to get a password prompt from the TACACS+ server. The operator enters a password, which the device relays to the server for validation. Upon successful authentication, the TACACS+ server supports both exec and command authorization similar to RADIUS authorization described above.

To use TACACS+ authentication, a Crypto-officer must configure TACACS+ server settings along with authentication and authorization settings.

Saving the configuration

After zeroizing, use the **write memory** command to save the configuration.

```
device(config)# write memory
```

Note
Keep a backup copy of the startup configuration in the event of system reset.

Reloading the device

After you have saved the configuration, reload the device using the **reload** command.

```
device# reload
```

Various tests, including Power-On Self-Test (POST), MACSec config integrity test (only when FIPS is enabled), and Known Answer Tests (KATs), are run by the Extreme device during reload, during the transition between non-FIPS mode and FIPS mode.

POST checks for the consistency of the FIPS-approved algorithms implemented on the device.

KATs are used to exercise various features of FIPS-approved algorithms.

All interfaces on the device are down until the tests are completed successfully.

Possible POST failure messages indicating that the tests did not pass successfully include the following messages:

```
Crypto module initialization and KNown Answer Test (KAT) failed with reason:(Error
Code 0x80000000)'CKR_VENDOR_DEFINED'

FIPS: Primary image verification failed

FIPS: Secondary image verification failed
```

If there is a failure while the POST is being run, the device will be restarted. Monitor mode can be accessed to troubleshoot the issue.

Note
Contact Extreme Technical Support if the error repeats again.

For information on access to monitor mode to perform debugging, refer to Access to monitor mode.

Use the **fips self-test** command to run tests on demand, in both FIPS mode and non-FIPS mode. Refer to Running FIPS self-test.

After all tests are completed successfully, the device reloads in FIPS mode and FIPS mode is successfully enabled and operational on the Extreme device.

You can verify the status of the device as operationally in FIPS mode by using the **fips show** command.

The following example shows **fips show** command output after the device reloads successfully in the default strict FIPS mode, and administrative status is on and operational status is on.

```
device# fips show
Cryptographic Module Version: EXTR-NI-IP-CRYPTO-VER-4.0
FIPS mode: Administrative status ON: Operational status ON
Common-Criteria: Administrative status OFF: Operational status OFF
System Specific
OS monitor access status is: Disabled

Management Protocol Specific:
Telnet server: Disabled
Telnet client: Disabled
TFTP client: Disabled
HTTPS SSL 3.0: Disabled
SNMP Access to security objects: Disabled

Critical security Parameter updates across FIPS boundary:
Protocol Shared secret and host passwords: Clear
Password Display: Disabled

HTTPS RSA Host Keys and Signature: Clear
SSH DSA Host keys: Clear
SSH RSA Host keys: Clear
```

The following example shows the output of the **fips show** command on a NetIron CER device, after the device reloads successfully in the default strict FIPS mode, and administrative status is on and operational status is on.

```
device(config)# fips show
FIPS Validated Cryptographic Module
MP FIPS Version: EXTR-NI-IP-CRYPTO-VER-4.0
LP FIPS Version: EXTR-NI-LP-CRYPTO-VER-2.0
FIPS mode : Administrative status ON: Operational status ON
FIPS CC mode: Administrative status OFF: Operational status OFF
System Specific:
OS monitor access status is: Disabled
Management Protocol Specific:
Telnet server : Disabled
Telnet client : Disabled
TFTP client : Disabled
HTTPS SSL 3.0 : Disabled
SNMP v1, v2, v2c : Disabled
SNMP Access to security objects: Disabled
Password Display : Disabled
Critical security Parameter updates across FIPS boundary:
(i.e. during "fips zeroize" ..., or "no fips enable") :
Protocol Shared secret and host passwords: Clear
SSH RSA Host keys : Clear
HTTPS RSA Host Keys and Signature : Clear
```

Performing a FIPS self-test

Use the FIPS self-test to verify the sanity of FIPS software.

About This Task

For more information on the FIPS self-test, refer to Running FIPS self-test.

Note
During FIPS self-test, the CPU usage is high. Use the **fips self-tests** command before the device is placed in FIPS operational or administrative modes. Execution of the **fips self-tests** command in FIPS operational or administrative modes may result in the device rebooting as per the FIPS criteria.

From the Privileged EXEC level of the CLI on the console, use the **fips self-tests** command to verify that the FIPS Software and Firmware Integrity Test passes.

The following example shows the FIPS Software and Firmware Integrity Test as passed:

```
device# fips self-tests
WARNING: Issuing of this command may result in your device reloading.
WARNING: Please verify firmware images are installed correctly first.
Are you sure? (enter 'y' or 'n'): y
fips crypto drbg health check tests ran successful.
FIPS Power On Self Tests and KAT tests successful.
Running FIPS Software/Firmware Integrity Test
Verifying MP Image file primary.....Verified OK
FIPS: Image verification passed for primary
PASSED
Verifying MP Monitor.....Verified OK
FIPS: Image verification passed for monitor
PASSED
Verifying LP Image file lp-primary-.....Verified OK
FIPS: Image verification passed for lp-primary-0
PASSED
Verifying LP Monitor.....Verified OK
FIPS: Image verification passed for lp-monitor-0
PASSED
FIPS Software/Firmware Integrity Test PASSED

Running continuous DRBG check.
Running continuous DRBG check successful.
Pairwise consistency check successful.

FIPS KAT and Conditional Tests... PASSED
```

If the test fails, make sure that the correct signature file was copied for the correct image file and version, and recopy as needed.

Note
The FIPS self-test must pass before saving the configuration and reloading the device.

Modifying the FIPS policy

After the device is administratively in FIPS mode, you can modify the default FIPS policy.

Note
Making changes to the default FIPS policy on the device is not recommended and weakens the security of the device. Any modification of the default FIPS policy places the device in a state that is not in compliance with FIPS 140-2.

The output of the **fips enable** command displays which protocols that constitute the FIPS policy are set in compliance with FIPS standards by default and can be adjusted to set a more flexible policy. The remaining protocols that constitute the FIPS policy are set to the appropriate status automatically during reload due to the **fips enable** command. The default FIPS policy is detailed in How FIPS works.

When you make no changes to the FIPS policy, the default FIPS policy is applied on the device and the device operates in strict FIPS mode upon reload, in full compliance with FIPS 140-2 specifications.

To set a more flexible FIPS policy on the Extreme device, use the following commands as desired to modify the default FIPS policy.

- Allow TFTP access:

```
device(config)# fips policy allow tftp-access
```

  Syntax: **[no] fips policy allow tftp-access**

- Allow SNMP access to the critical security parameter (CSP) MIB objects:

```
device(config)# fips policy allow snmp-csp-access
```

  Syntax: **[no] fips policy allow snmp-csp-access**

- Allow access to monitor mode for debugging both from application and boot prompts:

```
device(config)# fips policy allow monitor-full-access
```

  Syntax: **[no] fips policy allow monitor-full-access**

  Note
  During an application reset, monitor access is restored to allow debugging. Refer to Access to monitor mode.

  - Allow display of secrets and passwords in encrypted or clear text format:

```
device(config)# fips policy allow password-display
```

Syntax: **[no] fips policy allow password-display**

Note

In the FIPS default mode of operation, **enable password-display** cannot be configured. The various show commands will always mask the secret or password with ".....".

To override this behavior, the Crypto-officer can configure this policy, by using the **fips policy password-display** command, which allows **enable password-display** to be configured. The various show commands will display the secret or password in either encrypted or clear text form, depending on the implementation.

- Retain the shared secret keys for all protocols and the host passwords:

```
device(config)# fips policy retain shared-secrets
```

Syntax: **[no] fips policy retain shared-secrets**

- Retain the HTTPS RSA host keys and the HTTPS server digital certificate:

```
device(config)# fips policy retain rsa-host-keys
```

Syntax: **[no] fips policy retain rsa-host-keys**

Disabling FIPS mode

Use the **no fips enable** command to disable FIPS mode on the Extreme device.

After you enter the command, a warning displays that FIPS mode will be disabled.

This command performs the following policy-related operations:

- Enables TFTP access. Use followind command: **fips policy allow tftp-access**.
- Re-enables SNMP access to critical security parameter (CSP) MIB objects.
- Re-enables SNMPv3 encryption protocol DES for future SNMPv3 user configuration.
- Re-enables access to monitor mode.
- Zeroizes shared secrets, SSH and HTTPS host keys, and the HTTPS certificate based on the configured FIPS policy.

The **no fips enable** command also performs the non-policy-related operation of re-enabling the RC4 cipher for the HTTPS server.

Changes to the running configuration are not saved to the startup configuration; therefore, when the device reloads, it returns to FIPS mode.

Use the **write memory** command to save the running configuration.

Note

Use the **clear ikev2 sa** command to manually remove the connection once FIPS mode is disabled. You can use the **clear ikev2 sa** command after using the **fips zeroize all** command as well.

Running FIPS self-test

Use the **fips self-tests** command either in FIPS mode or non-FIPS mode to run the Known Answer Tests (KATs) and conditional tests on demand in both FIPS mode and non-FIPS mode.

```
device# fips self-tests
WARNING: Issuing of this command may result in your device reloading.
WARNING: Please verify firmware images are installed correctly first.
Are you sure? (enter 'y' or 'n'): y

fips crypto drbg health check tests ran successful.
FIPS Power On Self Tests and KAT tests successful.
Running FIPS Software/Firmware Integrity Test
Verifying MP Image file primary.....Verified OK
FIPS: Image verification passed for primary

SYSLOG: <14>Sep  6 19:31:53 FIPS: Image verification passed for primary
PASSED
Verifying MP Monitor.....Verified OK
FIPS: Image verification passed for monitor

SYSLOG: <14>Sep  6 19:32:04 FIPS: Image verification passed for monitor
PASSED
Verifying LP Image file lp-primary-.....Verified OK
FIPS: Image verification passed for lp-primary-0

SYSLOG: <14>Sep  6 19:32:05 FIPS: Image verification passed for lp-primary-0
PASSED
Verifying LP Monitor.....Verified OK
FIPS: Image verification passed for lp-monitor-0

SYSLOG: <14>Sep  6 19:32:16 FIPS: Image verification passed for lp-monitor-0
PASSED
FIPS Software/Firmware Integrity Test PASSED

Running continuous DRBG check.
Running continuous DRBG check successful.
Pairwise consistency check successful.

FIPS KAT and Conditional Tests... PASSED
```

Syntax: **fips self-tests**

The following log message is generated when the KAT is completed, but no trap messages are generated because the system is not fully operational.

```
“Crypto module initialization and Known Answer Test (KAT) passed”.
```

Access to monitor mode

The device in strict FIPS mode with the default policy applied does not allow access to monitor mode commands that perform memory access.

When the device is operating in FIPS mode, you can access all monitor mode commands, including memory debug commands, in the following instances:

- A flexible FIPS policy with the **fips policy allow monitor-full-access** command configured allows access memory debug commands.

- A strict FIPS policy does not allow access to memory debug commands. To apply a more flexible policy and allow access to all monitor commands, either configure a more flexible FIPS policy or disable FIPS mode to enter monitor mode. Refer to Accessing monitor mode from FIPS mode.

  Note
  Making changes to the default FIPS policy on the device is not recommended and weakens the security of the device.

- In the event of continuous reboot or failure on the Extreme device, you can access monitor mode to perform troubleshooting. Refer to Accessing monitor mode in the event of continuous failure.

Perform the necessary operations after allowing the device access to the memory debug commands. Refer to Debugging in monitor mode.

To enable FIPS mode on the device after you have completed your use of monitor mode, refer to Returning to FIPS mode from monitor mode.

Accessing monitor mode from FIPS mode

A flexible FIPS policy with the **fips policy allow monitor-full-access** command configured allows access to monitor mode memory debug commands.

About This Task

When the default FIPS policy is applied and the device is in strict FIPS mode, take the following steps to set a more flexible FIPS policy and allow access to debug commands.

Note
Making changes to the default FIPS policy on the device is not recommended and weakens the security of the device. Any modification of the default FIPS policy places the device in a state that is not in compliance with FIPS 140-2.

Procedure

1. Use the **fips zeroize all** command to clear the critical security parameters (CSPs). The device zeroizes the CSPs based on the configured FIPS zeroization policy.

   ```
   device(config)# fips zeroize all
   ```

2. Allow access to the restricted memory commands within monitor mode by using the **fips policy allow monitor-full-access** policy command.

   ```
   device(config)# fips policy allow monitor-full-access
   ```

   Syntax: **fips policy allow monitor-full-access**

What to Do Next

All commands in monitor mode, specifically the previously restricted memory access commands, are available for use. Refer to Debugging in FIPS mode.

If you do not want to apply any FIPS policy but the default and still need to enter monitor mode, disable FIPS mode on the device using the **no fips enable** command. Refer to Disabling FIPS mode.

Once FIPS is disabled, all monitor mode commands are available.

Accessing monitor mode in the event of continuous failure

In the event of continuous failure, enter monitor mode by pressing b during a boot cycle. Only a restricted CLI is available in monitor mode if the device was previously running in FIPS mode. This restricted CLI does not allow the use of commands that refer to the reading or writing memory location.

If you intend to run the memory access commands, erase the startup configuration file using the **`erase startup-config`** command. After the startup configuration is erased, the device lifts restrictions and starts with a blank configuration and FIPS mode is disabled. Use the **`reload`** command to reload the device. Refer to Reloading the device.

In this mode, you can download a new image to the device if required.

Debugging in monitor mode

After allowing access to monitor mode, the memory debug commands disabled in strict FIPS mode are available for use.

The monitor mode command set allows you to perform the following actions:

- Debug the system reset.
- Erase the configuration (reset CSPs).
- Set an IP address.
- Boot from TFTP.

Returning to FIPS mode from monitor mode

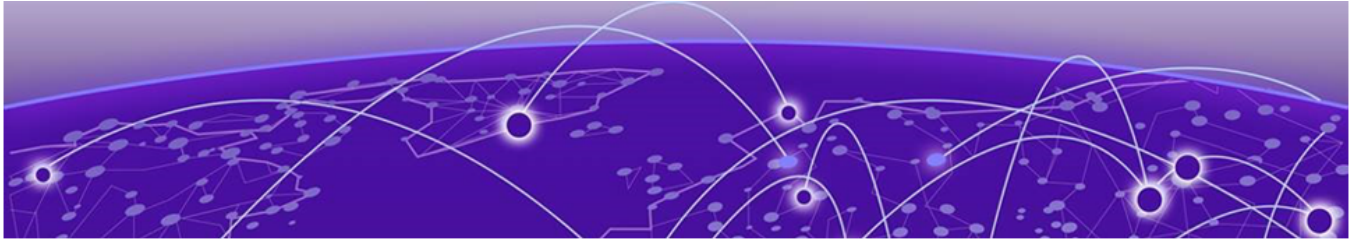
After the necessary actions are performed in monitor mode, take the following steps to return to FIPS mode.

Procedure

1. Use Ctrl + Z during reboot to exit monitor mode and return to the application prompt.
2. Re-create the CSP values.

What to Do Next

Use the **`fips enable`** command to re-enable FIPS mode on the device. Refer to Enabling FIPS mode.



Appendix: SP800-90A DRBG Implementation

DRBG support information on page 64

DRBG support information

Here is some additional information about the DRBG support implementation in Network OS.

1. There are no interfaces for external users to collect the DRBG generated by the crypto module. Applications that run as part of crypto module request and obtain the DRBG generated through library API calls. All the DRBG functions required for SP800-90A are invoked to generate and test the random bytes before providing it to the application.
2. Design of the implementation mandates validating every bit of the random value generated during the generation and timely re-seeding. Implementation also handles the un-instantiation to ensure that the residual values are not used for seeding.

   The implementation includes Health testing during all stages of DRBG generation: instantiate, seed, generate, reseed and un-instantiate.
3. The implementation utilizes CTR based DRBG mechanism with AES 256 cryptographic primitive for the generation of random numbers.
4. The implementation uses multiple entropy input sources to ensure that the entropy pool is full for generation of random bytes. In addition, the implementation always employs /dev/random to ensure the security strength of the entropy bits.
5. The implementation employs CTR-based DRBG mechanism with AES-256 cryptographic primitive with additional features to ensure stronger DRBG. Features included are predication resistance, additional input and personalization string.
6. DRBG mechanism functions are distributed in the implementation and hence no mechanisms are required to protect confidentiality and Integrity of the internal state.
7. The implementation uses CTR-based DRBG mechanism with derivation function.
8. In addition to the health test listed in SP800-90A, continuous random number generation tests are run on the bytes that are generated.
9. The DRBG health tests are run at an interval of every (1<<24) iterations of DRBG generation, which ensures that even the larger requirement for random numbers are validated.

   DRBG health tests are instantiated, seeded and generated for every requirement to generate the random number.

10. The DRBG functions can be tested in the implementation by power-cycle of the switch, key generation or any request for random numbers.
11. The SP800-90A DRBG implementation is part of the library whose installation is controlled within Extreme and can be downloaded on the crypto-module only through RSA 2048 and SHA256 verification.