BROCADE

FIPS CONFIGURATION GUIDE

Brocade NetIron FIPS and Common Criteria

Supporting NetIron R05.9.00aa

53-1003826-01
24 March 2017

Contents

Preface **7**
- Document conventions 7
  - Notes, cautions, and warnings 7
  - Text formatting conventions 7
  - Command syntax conventions 8
- Brocade resources 8
- Document feedback 8
- Contacting Brocade Technical Support 9
  - Brocade customers 9
  - Brocade OEM customers 9

About This Document **11**
- Supported hardware and software 11
- FIPS-supported devices 11
- FIPS-supported interface modules 12
  - Brocade MLXe platform 12
  - Brocade CER platform 13
  - Brocade CES platform 13

Federal Information Processing Standards **15**
- FIPS overview 15
- How FIPS works 16

Upgrading and Downgrading Software on FIPS-enabled Devices **19**
- Upgrading FIPS-enabled devices 19
  - Image verification in FIPS or CC mode 19
  - FIPS NetIron 5.9.00aa images for Brocade MLXe devices 19
  - Performing a basic upgrade 20
  - MACsec and software release upgrade 21
- Downgrading from FIPS mode to non-FIPS mode 21

FIPS Configuration **23**
- User roles in FIPS mode 23
- Commands disabled in FIPS mode 23
- Hidden files in FIPS mode 24
- Cryptographic algorithms in FIPS mode 24
  - Cryptographic algorithms on the management module 25
  - Cryptographic algorithms on the Brocade NetIron CES and CER devices 26
  - Cryptographic algorithms on the BR-MLX-10GX4-IPSEC-M module 26
  - Cryptographic algorithms on the BR-MLX-10GX20-M, BR-MLX-10GX20-X2, BR-MLX-1GX20-U10G-M, and BR-MLX-1GX20-U1 26
- SSH clients 27
- Usernames and SSH public key authentication 27
  - Implementation 27
  - Restrictions 27
- Protocol changes in FIPS mode 28
  - BGP 28
  - HTTP 29

HTTPS 29
IKEv2/IPsec 30
IS-IS 30
L2 over IPsec 30
MACsec 31
MPLS 31
NTP 31
OpenFlow 32
OSPFv2 32
OSPFv3 32
PKI 32
Proprietary 2-way encryption algorithms 33
RADIUS 33
SCP 33
SNMP 34
SSHv2 35
Syslog 36
TACACS+ 36
Telnet 37
TFTP 37
VRRP 37
VRRP-E 37
Web Authentication 37
DRBG Health Test on IPsec LP 37
System reset and boot up in FIPS mode 39
Debugging in FIPS mode 39
Placing the device in FIPS mode 39
General steps to place the Brocade NetIron device in FIPS mode 39
Copying the signature files 40
Enabling FIPS mode 42
Zeroizing shared secrets and host keys 46
Configuring user authentication 48
Saving the configuration 50
Reloading the device 50
Performing a FIPS self-test 52
Modifying the FIPS policy 52
Disabling FIPS mode 54
Running FIPS self-test 54
Access to monitor mode 55
Accessing monitor mode from FIPS mode 55
Accessing monitor mode in the event of continuous failure 56
Debugging in monitor mode 56
Returning to FIPS mode from monitor mode 56

Common Criteria Certification 57
Common Criteria overview 57
Features unavailable in Common Criteria mode 58
Enabling Common Criteria mode 58
Entering Common Criteria Administrative mode 59
Entering Common Criteria Operational mode 66
Displaying Common Criteria information 66

Encrypted syslog servers in Common Criteria mode....68
AAA servers in Common Criteria mode....68
Modifying the Common Criteria policies to use non-encrypted AAA servers....69
Downgrading from Common Criteria mode to non-FIPS mode....69
Commercial Solutions for Classified program....70
Network Device Protection Profile with VPN gateway....70
NDPP with VPN gateway requirements....70
NAT traversal in IKE and IPsec....71
Selecting the AES-GCM-128 algorithm....71
Audit logging....71
Configuring the strict password rules....72
Support for Logging IKE and PKI Transaction Details....72
Management commands....81

Configuring an Encrypted Syslog Server....85
Encrypted syslog server overview....85
Setting up stunnel....85
Creating a certificate with the OpenSSL toolkit....85
Creating a configuration file....85
Changing the stunnel4 startup file....86
Restarting the stunnel service....86
Configuring rsyslog....86
Enabling accepting remote logs....86
Restarting rsyslog service....87
Printing log messages....87
TLS encrypted syslog server configuration and validation....87

Syslog Messages....93
Syslog messages in FIPS mode....93

Running Tasks for Different NetIron Devices in FIPS Mode....99
MLXe tasks....99
CER, CER-4X, CER-RT-4X tasks....101

OpenSSL License....103
OpenSSL license overview....103
License....103

Preface

- Document conventions 7
- Brocade resources 8
- Document feedback 8
- Contacting Brocade Technical Support 9

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION
A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used to highlight specific words or phrases.

Format	Description
bold text	Identifies command names.
	Identifies keywords and operands.
	Identifies the names of GUI elements.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.
`Courier font`	Identifies CLI output.

Format	Description
	Identifies command syntax examples.

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	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. In Fibre Channel products, square brackets may be used instead for this purpose.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

White papers, data sheets, and the most recent versions of Brocade software and hardware manuals are available at www.brocade.com. Product documentation for all supported releases is available to registered users at MyBrocade.

Click the **Support** tab and select **Document Library** to access product documentation on MyBrocade or www.brocade.com. You can locate documentation by product or by operating system.

Release notes are bundled with software downloads on MyBrocade. Links to software downloads are available on the MyBrocade landing page and in the Document Library.

Document feedback

Quality is our first concern at Brocade, and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com
- By sending your feedback to documentation@brocade.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.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online or by telephone. Brocade OEM customers should contact their OEM/solution provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to www.brocade.com and select **Support**.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone
Preferred method of contact for non-urgent issues: • Case management through the MyBrocade portal. • Quick Access links to Knowledge Base, Community, Document Library, Software Downloads and Licensing tools	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • Toll-free numbers are available in many countries. • For areas unable to access a toll-free number: +1-408-333-6061

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/solution provider, contact your OEM/solution provider for all of your product support needs.

- OEM/solution providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/solution provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/solution provider.

About This Document

- Supported hardware and software....11
- FIPS-supported devices....11
- FIPS-supported interface modules....12

Supported hardware and software

The following hardware platforms are supported by FIPS:

- Brocade NetIron CES 2000-4X Series and Brocade NetIron CER 2000-4X-RT Series
- Brocade MLXe series (MLXe-4, MLXe-8, and MLXe-16) with management module (BR-MLX-MR2-M or BR-MLX-MR2-X)
- Brocade MLXe-32 with management module (BR-MLX-32-MR2-M or BR-MLX-32-MR2-X)

To determine if the Brocade device and current software version are FIPS-certified, refer to http://csrc.nist.gov/groups/STM/cmvp/validation.html.

To determine if the Brocade device and current software version is Common Criteria certified, refer to https://www.niap-ccevs.org/CCEVS_Products/pcl.cfm.

FIPS-supported devices

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

- Brocade MLXe-4, MLXe-8, and MLXe-16 with MR2 management modules (BR-MLX-MR2-M and BR-MLX-MR2-X)
- Brocade MLXe-32 with MR2 management module (BR-MLX-32-MR2-M or BR-MLX-32-MR2-X)
- Brocade MLXe with management module MR2: 1666 MHz Power PC processor 7448 (version 8004/0202) 166 MHZ bus
- Brocade CER/CES (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.

The following table lists the individual Brocade NetIron platforms that support FIPS-ready mode as detailed in FIPS Publication 140-2 requirements.

TABLE 1 Devices that support FIPS

Brocade NetIron XMR	Brocade MLX Series	Brocade MLXe Series with only MR2 management module (MLXe-4, MLXe-8, MLXe-16, and MLXe-32)	Brocade NetIron CER 2000 Series (4X-RT models only)	Brocade NetIron CES 2000 Series (4X models only)
No	No	Yes	Yes	Yes

NOTE

Beginning with Brocade NetIron 5.8.00, the NetIron devices do not support the MR management module.

FIPS-supported interface modules

FIPS is vendor-ready for the following interface modules:

- BR-MLX-10GX20-M
- BR-MLX-10Gx20-X2
- BR-MLX-1GX20-U10G-M
- BR-MLX-1GX20-U10G-X2
- BR-MLX-10GX4-IPSEC-M

Brocade MLXe platform

The following Brocade MLXe management modules are supported for certification:

- BR-MLX-MR2-M
- BR-MLX-MR2-X
- BR-MLX-32-MR2-M
- BR-MLX-32-MR2-X

The following Brocade MLXe chassis bundles are supported for certification:

- BR-MLXE-4-MR2-M-AC
- BR-MLXE-4-MR2-M-DC
- BR-MLXE-4-MR2-X-AC
- BR-MLXE-4-MR2-X-DC
- BR-MLXE-8-MR2-M-AC
- BR-MLXE-8-MR2-M-DC
- BR-MLXE-8-MR2-X-AC
- BR-MLXE-8-MR2-X-DC
- BR-MLXE-16-MR2-M-AC
- BR-MLXE-16-MR2-M-DC
- BR-MLXE-16-MR2-X-AC
- BR-MLXE-16-MR2-X-DC
- BR-MLXE-32-MR2-M-AC
- BR-MLXE-32-MR2-M-DC
- BR-MLXE-32-MR2-X-AC
- BR-MLXE-32-MR2-X-DC

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

- NI-X-4-HSF
- NI-X-16-8-HSF
- NI-X-32-HSF

Brocade CER platform

The following Brocade CER chassis bundles are supported for certification:

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

Brocade CES platform

The following Brocade CES chassis bundles are supported for certification:

- BR-CES-2024C-4X-AC
- BR-CES-2024C-4X-DC
- BR-CES-2024F-4X-AC
- BR-CES-2024F-4X-DC

NOTE

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

Federal Information Processing Standards

- FIPS overview..... 15
- How FIPS works.....16

FIPS overview

A Brocade 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.

NOTE

To determine if the NetIron device and current software version are FIPS-certified, refer to http://csrc.nist.gov/groups/STM/cmvp/validation.html.

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

A Brocade 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 my.brocade.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 administrative 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 on page 52.

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 v1.0/1.1 and TLS v1.2.

NOTE
Using **openflow enable** and **copy https** commands violates the Security Policy of the module and it deems the module non-compliant in the non-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.
- SNMP access to CSP 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

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.

Upgrading and Downgrading Software on FIPS-enabled Devices

- Upgrading FIPS-enabled devices 19
- Downgrading from FIPS mode to non-FIPS mode 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 Brocade Technical Support and perform the procedure under qualified guidance.

Image verification in FIPS or CC 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 Brocade 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 5.9.00aa images for Brocade 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 Brocade technical support.

TABLE 2 Required images for a basic upgrade to NetIron 5.9.00aa

Image description	Image name	Signature name for upgrade from devices running (legacy) NetIron 5.7.00a and earlier code using DSA1024/SHA1 signatures	RSA2048/SHA256 bit signatures file name
Combined application image for management modules	xm05900.bin	xm05900.sig	xm05900.sha256

TABLE 2 Required images for a basic upgrade to NetIron 5.9.00aa (continued)

Image description	Image name	Signature name for upgrade from devices running (legacy) NetIron 5.7.00a and earlier code using DSA1024/SHA1 signatures	RSA2048/SHA256 bit signatures file name
Monitor image for management modules	xmb05900.bin	xmb05900.sig	xmb05900.sha256
Monitor image for interface modules	xmlb05900.bin	xmlb05900.sig	xmlb05900.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	lpfpga05900.bin	lpfpga05900.sig	lpfpga05900.sha256

Performing a basic upgrade

The overall procedure for a basic upgrade involves copying only the new application, boot, monitor, and combined FPGA images. For more information on how to upgrade additional images, refer to the *Brocade NetIron Software Upgrade Guide*.

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

- Using Secure Copy (SCP). For more information about SCP, refer to the Brocade 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"
!
```

The output of the **show dot1x-mka config** command provides information about the bypass status for every port.

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

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 Brocade 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.

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 common-criteria** 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.

FIPS Configuration

- User roles in FIPS mode 23
- Commands disabled in FIPS mode 23
- Hidden files in FIPS mode 24
- Cryptographic algorithms in FIPS mode 24
- SSH clients 27
- Usernames and SSH public key authentication 27
- Protocol changes in FIPS mode 28
- DRBG Health Test on IPsec LP 37
- System reset and boot up in FIPS mode 39
- Debugging in FIPS mode 39
- Placing the device in FIPS mode 39
- Disabling FIPS mode 54
- Running FIPS self-test 54
- Access to monitor mode 55

User roles in FIPS mode

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

A Brocade 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 Brocade 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 Brocade 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 on page 25
- Cryptographic algorithms on the Brocade NetIron CES and CER devices on page 26
- Cryptographic algorithms on the BR-MLX-10GX4-IPSEC-M module on page 26
- Cryptographic algorithms on the BR-MLX-10GX20-M, BR-MLX-10GX20-X2, BR-MLX-1GX20-U10G-M, and BR-MLX-1GX20-U10G-X2 modules on page 26

Allowed exceptions include:

- RSA Key Wrapping
- Diffie-Hellman (DH)
- Elliptic curve Diffie-Hellman (ECDH)
- 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 v1.0/1.1 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
- Diffie-Hellman (DH)
- 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 Brocade NetIron CES and CER devices

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

- SNMPv3 (in authPriv security mode) KDF SP800-135
- TLS 1.2 KDF SP800-135
- Advanced Encryption Algorithm (AES) including AES-CTR and AES-128-CFB128
- Secure Hash Algorithm (this includes all SHA variants the module supports: SHA-1, SHA-256, SHA-384, and SHA-512)
- Keyed-Hash Message Authentication code (HMAC-SHA1, HMAC-SHA256)
- Deterministic Random Bit Generator (DRBG) Hash based
- Reversible Digital Signature Algorithm (RSA) including RSA2
- SSH Key exchange algorithm diffie-hellman-group-exchange-sha256

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

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

- IKEv2 KDF SP800-135
- ECDSA
- KAS ECC SP800-56A
- KAS FFC SP800-56A
- DRBG SP800-90A
- AES (AES-256-ECB)
- GCM (SP800-38D)
- Elliptical Curve Diffie-Hellman (ECDH)

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)
- HMAC (HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512)

Algorithms running on the PHY crypto engine:

- AES (AES-128-GCM)

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

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

- AES (AES-128-GCM)

SSH clients

SSH clients must be FIPS 186-3-compliant. You can use the OpenSSH-based client that is developed by Brocade to be FIPS 186-3-compliant.

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-brocade-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-brocade-privilege-level) as shown in the public key example in Implementation on page 27.

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 3 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	Yes	Yes	BGP on page 28
HTTP	No	Yes	HTTP on page 29
HTTPS	Yes, with limitations	Yes	HTTPS on page 29
IPsec	Yes, with limitations	Yes	IKEv2/IPsec on page 30
IS-IS	Yes	Yes	IS-IS on page 30
MACsec	Yes	Yes	MACsec on page 31
MPLS	Yes	Yes	MPLS on page 31
NTP	Yes, with limitations	Yes	NTP on page 31
OpenFlow	No	Yes	OpenFlow on page 32
OSPFv2	Yes	Yes	OSPFv2 on page 32
OSPFv3	Yes	Yes	OSPFv3 on page 32
PKI	Yes	Yes	PKI on page 32
Proprietary 2-way encryption algorithms	No	Yes	Proprietary 2-way encryption algorithms on page 33
RADIUS	Yes, with limitations	Yes	RADIUS on page 33
SCP	Yes	Yes	SCP on page 33
SNMP	Yes, with limitations	Yes	SNMP on page 34
SSHv2	Yes, with limitations	Yes	SSHv2 on page 35
Telnet	No	Yes	Telnet on page 37
TACACS+	Yes, with limitations	Yes	TACACS+ on page 36
TFTP	No	Yes	TFTP on page 37
Web Authentication	No	Yes	Web Authentication on page 37

NOTE

For more information on RADIUS authentication commands, refer to the *Brocade NetIron Command Reference* and the *Brocade NetIron 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 *Brocade 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 on page 33.
- 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 on page 33.
- SSL version 3 and earlier versions are disabled and TLS 1.0 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** { 0 | 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 TLS_RSA_WITH_AES_256_CBC_SHA 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.

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 Authentication Key
- IKEv2 KDF State
- IKEv2 Pre-Shared Key (PSK)

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 *Brocade 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 Brocade 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 *Brocade* NetIron *Routing Configuration Guide*.

NTP

Brocade 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 non-FIPS mode and not supported in the FIPS mode.

In the FIPS approved mode of operation, though this feature is available but it cannot be configured to be functional.

Using the **openflow enable** and **copy https** commands violates the Security Policy of the module and it deems the module non-compliant in the non-FIPS mode.

NOTE
To ensure that this feature is disabled you can run the **no openflow enable** command.

OSPFv2

The OSPFv2 protocol uses MD5 for authentication.

NOTE
OSPFv2 is not allowed in the FIPS approved mode as per the Security Policy.

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 *Brocade NetIron Routing Configuration Guide*.

OSPFv3

The OSPFv3 protocol uses IPsec with ESP and HMAC-SHA1-96 for authentication. HMAC-SHA-1-96 is a FIPS 140-2 Approved security function.

NOTE
OSPFv3 is not allowed in the FIPS approved mode as per the Security Policy.

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 1234567890abcde12345098765432112345678900
```

For more information on OSPFv3 authentication commands, refer to the *Brocade 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

For more information about PKI and IKEv2, refer to the specific sections in the *Brocade 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. To authorize an authentication, use commands such as the following to configure shared secret keys for RADIUS:

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

- RADIUS Secret

  NOTE

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

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 4 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 *Brocade 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 *Brocade 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 brocadeSNMP
```

SNMP notification

In the FIPS mode or CC mode of operation, the Brocade 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 Brocade 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 *Brocade 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 *Brocade NetIron Security Configuration Guide*.

The following table shows the supported SSH ciphers.

TABLE 5 SSH ciphers supported by NetIron devices

Brocade 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 Brocade 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 *Brocade* 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 on page 33.

- **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**. Brocade 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**. Brocade devices support plaintext and HMAC MD-5 authentication.

Web Authentication

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

DRBG Health Test on IPsec LP

Deterministic Random Bit Generator (DRBG) health and error checks are performed on the IPsec line card used in MLXe.

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 CLI 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 (e.g., AES-128 or SHA-256), are as follows:

- Type of DRBG mechanism: Hash Based
- Cryptographic primitives used: SHA-256

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

The implementation of this feature (e.g., prediction resistance, personalization string, additional input) are as follows:

- Prediction Resistance is not TRUE.
- Personalization String Length = 0
- Additional Input Length = 0

NOTE

The DRBG mechanism functions are not distributed. `CTR_DRBG` is not used. The code used to perform the DRBG Health Test on IPSec line card is from OpenSSL FIPS2.05.

Example CLI

DRBG functions can be tested on a demand basis, using CLI commands, by independent requests as shown in the following CLI example.

```
fips crypto drbg
LP-1#fips crypto drbg
Initializing Hash based sha-256 drng
Instantiating drbg
Running self tests on drbg
                DRBG SHA256 test started
                DRBG SHA256 test OK
FIPS CRYPTO: DRBG test PASSED
LP-1#fips crypto force-failure drbg
LP-1#fips crypto drbg
Initializing Hash based sha-256 drng
Instantiating drbg
Running self tests on drbg
                DRBG SHA256 test failure induced
                DRBG SHA256 test failed as expected
FIPS CRYPTO: DRBG test failed as expected
Aug 20 13:36:15:C:System: Module in slot 1 is rebooted due to FIPS DRBG KAT failure
Aug 20 13:36:15:N:Module 1 is reset by mgmt (reason: FIPS KAT failure)
Module is dow
NetIron XMR/MLX Boot Monitor Version 5.9.0
Enter 'b' to stop at boot monitor
sent IPC_MSGTYPE_REBOOT to slot 16 (my_slot = 0, ipc_post_rx32_mode = 0)
received IPC_MSGTYPE_REBOOT_ACK from fid d020
get_module_type: board_class = 244
```

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 or auxiliary flash card 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 on page 52.
- 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 on page 56.
- 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 on page 55.

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 Brocade NetIron device in FIPS mode

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

1. Assume the Crypto-officer role.
2. Copy the needed signature files. Refer to Copying the signature files on page 40.
3. Enable FIPS mode. Refer to Enabling FIPS mode on page 42. 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 on page 46.
5. Configure all users of the module and the authentication methods. Refer to Configuring user authentication on page 48
6. Save the configuration. Refer to Saving the configuration on page 50.
7. Reload the device. Refer to Reloading the device on page 50.
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 Performing a FIPS self-test on page 52.

10. Inspect the physical security of the module including placement of tamper evident labels on the Brocade NetIron device. Refer to the *Brocade 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 *Brocade 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 Brocade MLXe 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 6 Required signature files for the Brocade 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 7 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 Brocade CES/CER 2000-4X devices

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 Brocade NetIron MLXe devices

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.

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 Brocade 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:

   ```
   device# fips show
   FIPS Validated Cryptographic Module
   MP FIPS Version: BRCD-IP-CRYPTO-VER-3.0a
   LP FIPS Version: BRCD-LP-CRYPTO-VER-1.0a
   FIPS mode   : Administrative status OFF: Operational status OFF
   FIPS CC mode: Administrative status OFF: Operational status OFF

   device(config)# fips show
   FIPS mode: Administrative Status: OFF, Operational Status: OFF
   ```

   If the device is already in administrative FIPS mode, you can modify the FIPS policy. Refer to Modifying the FIPS policy on page 52.

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 Brocade MLX Series and Brocade NetIron XMR 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, Brocade 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. 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 will be disabled.
      -TLS version 1.0 and greater will be used.
      -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 Brocade 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, Brocade 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 Brocade 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: BRCD-IP-CRYPTO-VER-3.0a
LP FIPS Version: BRCD-LP-CRYPTO-VER-1.0a
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 a Brocade 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
FIPS Version: BRCD-IP-CRYPTO-VER-3.0a
FIPS mode   : Administrative status ON: Operational status ON
FIPS CC mode: Administrative status OFF: Operational status OFF

System Specific:
OS monitor access status is: Enabled

Management Protocol Specific:
Telnet server                  : Disabled
Telnet client                  : Disabled
TFTP client                    : Enabled
SNMP v1, v2, v2c               : Disabled
SNMP Access to security objects: Enabled
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                        : Retain
```

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.

```
Brocade# 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.

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
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 tables list the various keys used in the system that are zeroized in compliance with FIPS.

TABLE 8 Key zeroization

Keys used	Type
IKEv2 DH Group-14 Private Key 2048 bit MODP	Host-keys
IKEv2 DH Group-14 Public Key 2048 bit MODP	Host-keys
IKEv2 DH Group-14 Shared Secret 2048 bit MODP	Shared-secret
IKEv2 ECDH Group-19 Private Key (P-256)	Host-keys
IKEv2 ECDH Group-19 Public Key (P-256)	Host-keys
IKEv2 ECDH Group-19 Shared Secret (P-256)	Shared-secret
IKEv2 ECDH Group-20 Private Key (P-384)	Host-keys
IKEv2 ECDH Group-20 Public Key (P-384)	Host-keys
IKEv2 ECDH Group-20 Shared Secret (P-384)	Shared-secret
IKEv2 ECDSA Private Key (P-256)	Host-keys
IKEv2 ECDSA Private Key (P-384)	Host-keys
IKEv2 ECDSA Public Key (P-256)	Host-keys
IKEv2 ECDSA Public Key (P-384)	Host-keys
IKEv2 Encrypt/Decrypt Key	NA
IKEv2 KDF State	Shared-secret
IKEv2 Pre-Shared Key (PSK)	Shared-secret
IKEv2/IPSec Authentication Key	Shared-secret
IPsec ESP Encrypt/Decrypt Key	NA
Local - Crypto-officer Password	NA
Local - Port Administrator Password	NA
Local - User Password	NA
LP DRBG Internal State	NA
LP DRBG Seed	NA
LP DRBG Value C	NA
LP DRBG Value V	NA
MKA Connectivity Association Key (CAK)	Shared-secret

TABLE 8 Key zeroization (continued)

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

Configuring user authentication

Brocade 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.

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.

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 Brocade 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 Brocade Technical Support if the error repeats again.

For information on access to monitor mode to perform debugging, refer to Access to monitor mode on page 55.

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 on page 54.

After all tests are completed successfully, the device reloads in FIPS mode and FIPS mode is successfully enabled and operational on the Brocade 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: BRCD-IP-CRYPTO-VER-3.0
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: BRCD-IP-CRYPTO-VER-3.0
LP FIPS Version: BRCD-LP-CRYPTO-VER-1.0
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.

For more information on the FIPS self-test, refer to Running FIPS self-test on page 54.

> 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 on page 16.

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 Brocade 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 on page 55.

– 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 Brocade 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.
- 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
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
```

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 on page 55.

  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 Brocade device, you can access monitor mode to perform troubleshooting. Refer to Accessing monitor mode in the event of continuous failure on page 56.

Perform the necessary operations after allowing the device access to the memory debug commands. Refer to Debugging in monitor mode on page 56.

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 on page 56.

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.

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.

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

All commands in monitor mode, specifically the previously restricted memory access commands, are available for use. Refer to Debugging in FIPS mode on page 39.

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 on page 54.

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 on page 50.

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.

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

Use the **fips enable** command to re-enable FIPS mode on the device. Refer to Enabling FIPS mode on page 42

Common Criteria Certification

- Common Criteria overview 57
- Enabling Common Criteria mode 58
- Encrypted syslog servers in Common Criteria mode 68
- AAA servers in Common Criteria mode 68
- Downgrading from Common Criteria mode to non-FIPS mode 69
- Commercial Solutions for Classified program 70
- Network Device Protection Profile with VPN gateway 70

Common Criteria overview

Common Criteria certification for a device enforces a set of security standards and feature limitations on a device to be compliant with the Common Criteria standards, similar to placing the device in FIPS mode. These restrictions are in addition to the requirements of FIPS mode. When the device is placed in Common Criteria mode, several security features that are available in FIPS mode are unavailable on the device. Because Common Criteria mode enforces security restrictions additional to FIPS mode, procedures and information are provided in relation to those for the FIPS mode.

For information about enabling FIPS mode on the device, refer to FIPS Configuration on page 23.

NOTE
Common Criteria mode becomes available once a device is FIPS-enabled.

NOTE
To determine if the NetIron device and current software version is Common Criteria-certified, refer to https://www.niap-ccevs.org/CCEVS_Products/pcl.cfm. Refer to the release notes for the software version running on the device to verify that the software is FIPS-and Common Criteria-certified.

You can enable Common Criteria mode on a device directly from non-FIPS mode, or on a device already in FIPS mode. The following table summarizes the transitions.

TABLE 9 Transition to Common Criteria mode

From	To non-FIPS mode	To FIPS mode	To Common Criteria mode
Non-FIPS mode	Not applicable	Use the **fips enable** command	Use the **fips enable common-criteria** command
FIPS mode	Use the **no fips enable** command	Not applicable	Use the **fips enable common-criteria** command
Common Criteria mode	Use the **no fips enable** or **no fips enable common-criteria** command	Use the following commands in a sequence: 1. **no fips enable** 2. **reload device** 3. **fips enable**	Not applicable

Be advised of the following considerations:

- Disabling FIPS mode from the Common Criteria mode using the **no fips enable** command downgrades the device directly into the non-FIPS mode.

- You cannot directly transition from Common Criteria mode to FIPS mode. To transition to FIPS mode, you must disable FIPS mode, reload the device, and then enable FIPS mode.

The following table lists the individual Brocade NetIron platforms that support Common Criteria certification requirements.

TABLE 10 Devices that support Common Criteria

Features supported	Brocade NetIron XMR	Brocade MLX Series	Brocade NetIron CES 2000 Series BASE package	Brocade NetIron CES 2000 Series ME_PREM package	Brocade NetIron CES 2000 Series L3_PREM package	Brocade NetIron CER 2000 Series Base package	Brocade NetIron CER 2000 Series Advanced Services package
FIPS CC mode	No	MLXe: Yes MLX: No	No	No	No	No	Yes

NOTE

Only the MLXe-4, MLXe-8, and MLXe-16, and CER 2000 Advanced Services Package are FIPS Common Criteria-certified.

Features unavailable in Common Criteria mode

Some of the security features that are allowed in FIPS mode are disabled in Common Criteria mode:

- SSHv2: Host and client key generation methods using DSA and the RSA-1024 key size are not supported (only RSA 2048 and higher key sizes are supported). Therefore, the following commands are not supported:
  - **crypto key generation dsa**
  - **crypto key client generation dsa**
  - **crypto key zero dsa**
  - **crypto key client zero dsa**
  - **crypto key gen rsa modulus 1024**
  - **crypto key zero rsa modulus 1024**
- TLS and HTTPS: The RSA 1024 key size for SSL or TLS private key generation is not supported (NetIron devices support only 2048 and above key sizes).
- SSH key exchange: The SSH key exchange method Diffie-Hellman-Group1-Sha1 is not supported. Only Diffie-Hellman-Group14-Sha1 is supported.
- Syslog: Logging to a host that uses UDP for transport is not supported. Only the TLS host is supported. Therefore, the **logging host** [**ipv4** | **ipv6**] {*ip-address* | *ipv6-address*} **ssl-port** *port* command is not supported. Refer to Configuring an encrypted syslog server for more information.
- AAA servers: Only local and TLS-encrypted TACACS+ servers are supported.

Enabling Common Criteria mode

When you enable Common Criteria mode on the device, it enters the Common Criteria Administrative mode. Similar to FIPS, Common Criteria also has administrative and operational modes:

- Common Criteria Administrative mode: Log in to the device console and enable the Common Criteria mode. You can optionally modify the default Common Criteria security policy in this mode.

NOTE
When you use the **reload** command to reload the device, the validation of software image with the signature file is triggered. Failure in signature verification results in the device continuously rebooting after device reload.

- Common Criteria Operational mode: Transition to Common Criteria operational mode from Common Criteria Administrative mode. After you transition the device to the Operational mode, you must save the configuration and reboot the device.

  NOTE
  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.

Entering Common Criteria Administrative mode

You can enable Common Criteria mode on a device with the following command.

```
device(config)# fips enable common-criteria
```

Syntax: [no] fips enable common-criteria

The device prompt displays the detailed banner information shown in the following example.

```
device(config)# fips enable common-criteria
WARNING: This will enable FIPS and Common Criteria 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 CC administrative mode.
At this time you can alter this system's CC default security policy
and then enter CC operational mode.

Note: Making changes to the default CC security policy weakens
the security of the device and makes the device non-compliant
with CC and 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, Brocade does not recommend
making changes to the default security policy at any time.
====================================

To enter CC 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 CC mode:
FIPS: SCP is already enabled
      - 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.
====================================

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

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.
NTP md5 authentication will be disabled.
HTTP Client will be disabled.
Passwords/Keys which dont comply FIPS standards will be removed on reload.
Please see FIPS config guide for complete details.

====================================
Additionally, in CC operational mode, following are the restrictions
on system services or commands after reload:
CC. Syslog servers need to use TLS encapsulation(see exception below in VPNGW).
CC. TACACS+ servers need to use TLS encapsulation(see exception below in VPNGW).
CC. DSA keys will be deleted from configuration, and will be disabled .
CC. RSA key sizes will be restricted to 2048 and above in the configuration.
CC. RADIUS servers should not be used (see exception below in VPNGW).
CC. For SSH Key Exchange, only diffie-hellman-group-exchange-sha256
    algorithm is allowed.

In CC VPN Gateway mode, since all communication happens over IPSec
using the out-of-band ports, here are the guidelines:
VPNGW. Management port should not be used since management module does
       not have IPSec stack
VPNGW. Syslog servers could be configured to use UDP. No need to use
       TLS encapsulation
VPNGW. TACACS+ servers could be configured to use TCP. No need to use
       TLS encapsulation
VPNGW. RADIUS servers could be configured to use UDP. No need to use
       TLS encapsulation
VPNGW. The required logging needs to be separately enabled:
       "log enable ikev2-extended"
       "log enable pki-extended"
VPNGW. The NAT-T needs to be separately enabled:
       "ikev2 nat-enable"
      - Start SSL client task for secure syslog server.
          Current status: Enabled
      - Configuration "enable aaa console" will be removed.
          Current status is: Disabled
```

General considerations when the device is in the Common Criteria Administrative mode

The following general considerations apply when the device is in the Common Criteria administrative mode on the MP (applies only to the VPNGW mode and IPsec must be used for VPNGW).

- Use RADIUS/UDP over the IPsec tunnel configured for managing the device.
- IPsec stack is not available on the management port.
- Configure the VPN gateway separately since it requires logging into the device.
- Configure the VPN gateway NAT translation separately.
- VPN gateway allows TACACS+ to use IPsec instead of TLS.
- VPN gateway allows Syslog to use IPsec instead of TLS.
- The extended IKEv2 and the extended PKI logging needs to be enabled to log the entire contents of packets associated with establishing a session with an IPsec peer.

Configuring IPv4 and IPv6 IPsec Tunnels

You configure IPsec when you want to set up an IPsec IPv4 or IPv6 tunnel to use to transmit secured IP packets. The configuration involves specifying the tunnel interface using the tunnel number, the tunnel mode (IPv4 or IPv6), and the IKE and IPsec options for the tunnel. Once the tunnel is created, you can use it to transmit IP packets.

You can use an IPsec IPv6 tunnel to transmit IPv4 IP packets, but you cannot use an IPsec IPv4 tunnel to transmit IPv6 IP packets.

> NOTE
> The network administrator must ensure that the IKE cryptographic algorithms and key sizes that are configured for a tunnel are not stronger than the IPsec cryptographic algorithms and key sizes used by the same tunnel.

Affect of authentication method on IKEv2 profile settings

The type or method of authentication you select for IKE transactions affects IKEv2 profile options you should select when setting up IPsec tunnels.

The recommended IKEv2 profile options are:

- **PKI-based authentication** When using PKI-based authentication, it is recommended that you select Distinguished Name (DN).
- **Pre-shared key authentication** When using pre-shared key authentication (PSK), it is recommended that you select Fully Qualified Domain Name (FQDN).

Format requirements for text-based PSK

Text-based PSK can be up to 100 characters in length, and all characters must be from any of the following sets:

- **Lower case letters:** a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z
- **Upper case letters:** A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
- **Number:** 1,2,3,4,5,6,7,8,9,0
- **Special Characters:** “!”,”@”,”#”,”$”,”%”,”^”,”&”,”*”,”(“,”)”
- **Additional special characters:** “”, “{“,”}”,”[“,”]”,”-“,”_”,”=”,”+”,”;”,”:”,”,”,”.”,”<”,”>”,”/”,”\?”,”|”,`”,

  (To use “?” it must be preceded by “\” (for example, “\?”.)

Format requirements for hex-based PSK

Hex-based PSK can be can be maximum length of 100 hex digits from the set below:

- 1,2,3,4,5,6,7,8,9,0,a,b,c,d,e,f,A,B,C,D,E,F.

Selecting text-based or hex-based PSK

You do not have to use any parameters to select text-based PSK. When using text-based PSK, you only have to enter the value for the key.

To select hex-based PSK, you must use the keyword **hex-based**, and enter the value for the key as hex digits.

Configuration examples

The following examples show configuration of text-based PSK and hex-based PSK.

Text-based PSK

This example shows a text-based PSK configuration. In this example, **psk-example1** is the name of the IKEv2 authentication proposal being configured for text-based PSK.

```
device(config)#
device(config)#ikev2 auth-proposal psk-example1
device(config-ike-auth-psk-example1)#pre-shared-key ?
  ASCII string   specifies the pre-share-key,maximum 100 characters
  hex-based      specifies hex based pre-share-key
device(config-ike-auth-psk-example1)#pre-shared-key
abcdefghijkmnopqrstuvwxyz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ!@#$%^&*()
```

Hex-based PSK

This example shows a hex-based PSK configuration. In this example, **psk-example2** is the name of the IKEv2 authentication proposal being configured for hex-based PSK. The **hex-based** keyword is used to specify hex-based PSK.

```
device(config)#
device(config)#ikev2 auth-proposal psk-example2
device(config-ike-auth-psk-example2)#pre-shared-key ?
  ASCII string   specifies the pre-share-key,maximum 100 characters
  hex-based      specifies hex based pre-share-key
device(config-ike-auth-psk-example2)#pre-shared-key hex-based ?
  HHHHHHHHHH    specifies the pre-share-key, maximum 100 hex digits
  <cr>
device(config-ike-auth-psk-example2)#pre-shared-key hex-based 1234567890abcdef0987654321ABCDEF
```

Pre-requisites:

- **Use of PKI**: If your IPsec tunnel configuration involves the use of PKI options (for example, PKI entity or PKI trust point), make sure you complete the PKI configuration before you begin setting up the IPsec tunnel. The PKI options must be configured before you can select them as part of the tunnel setup process. (See *Configuring PKI* in the Brocade NetIron Security Guide for descriptions of the PKI elements and how to configure them.)
- **Use of global IKEv2 parameters**: If you need to configure any global IKEv2 parameters (such as NAT-Traversal), make sure you complete the configuration before you begin setting up the IPsec tunnel. (See *Using Unicast IPSec IPv4 with Network Address Translation (NAT)* in the Brocade NetIron Security Guide for details.)
- **Use of AES-GCM-128**: If you choose to use the AES-GCM-128 algorithm for encryption and decryption of IP packets transmitted across the tunnel, make sure that:
  - The tunnel end nodes (local and remote) both use NetIron 5.9.0a.
  - The remote device has consistent configuration parameter settings for the tunnel.

Complete the following steps to set up the IPsec tunnel.

1. (Optional) Enter one of the following command on the management module (MP) to generate key-pairs using ECDSA P-384 or P-256 for signature generation and verification. (More than one key-pair can be generated with different label names.)

   ```
   device(config)# crypto key generate ec label <label-name> size 384
   device(config)# crypto key generate ec label <label-name> size 256
   ```

2. (Optional) Enter the following command on the management module (MP) to configure the PKI trustpoint to use the keys generated using ECDSA. Make sure you specify the same label name used in the previous step.

   ```
   device(config)# eckeypair key-label <label-name>
   ```

3. (Optional) Enter the following command to generate a certificate request (the request is sent to the trust point you specify by name). You must specify the trust point name.

   ```
   device(config)# pki enroll <name>
   ```

NOTE
Steps 1, 2, and 3 are applicable only when we are using the dynamic PKI. They are not applicable when using manual PKI or PSK-based authentication.

4. (Optional) Import the required certificates from the flash memory using the **pki import** *name* **pem url flash:** *file-name password* command and authenticate the trustpoint using the **pki authenticate** *trustpoint-name* command.

   NOTE
   Step 4 is applicable only when we are using the manual PKI. It is not applicable when using the dynamic PKI or PSK-based authentication.

5. Enter one of the following commands in tunnel interface configuration mode to select the tunnel mode for the IPsec VTI.
   - IPv4: **tunnel mode ipsec ipv4**
   - IPv6: **tunnel mode ipsec ipv6**

   ```
   device(config)# interface tunnel 1
   device(config-tnif-1)# tunnel mode ipsec ipv4
   ```

6. Enter the **tunnel source** command to specify the tunnel source. This is the local endpoint of the tunnel.

   The tunnel source can be one of the following:
   - The IPv4 address of a physical, virtual, or loopback interface.

     ```
     device(config) interface tunnel 1
     device(config-tnif-1)# tunnel source 192.168.1.2
     ```

   - The global IPv6 address of a physical, virtual, or loopback interface.

     ```
     device(config) interface tunnel 1
     device(config-tnif-1)# tunnel source 10:1:1::1/64
     ```

   - The interface on which the required tunnel source IPv4 address or IPv6 address has been configured.

     ```
     device(config) interface tunnel 1
     device(config-tnif-1)# tunnel source ethernet 1/1
     ```

7. Enter the **tunnel destination** command to specify the tunnel destination. This is the remote endpoint of the tunnel.

   ```
   device(config) interface tunnel 1
   device(config-tnif-1)# tunnel destination 10:1:1::2/64
   ```

8. Enter the **ip address** command to specify the IP address of the tunnel. This is the IP address of the tunnel port, not the IP address of a tunnel endpoint.

   ```
   device(config)# interface ethernet 3/1
   device(config-int-e10000-3/1)# ip address 36.0.8.108/32
   ```

9. Enter the **ipsec profile** command at the IPsec configuration level to add the IPsec parameters used between two IPsec-enabled Brocade devices, and enter IPsec profile configuration mode.
   In this example, an IPsec profile named test-profile has been created.

   ```
   device(config-ipsec)# ipsec profile test-profile
   device(config-ipsec-profile)#
   ```

10. Enter the **tunnel protection ipsec profile** command in tunnel interface configuration mode to configure an IPsec profile used to encapsulate the outgoing packets. (This binds the profile to the VTI.)

```
device(config)# interface tunnel 1
device(config-tnif-1)# tunnel protection ipsec profile test-profile
```

NOTE

The remaining steps are used to select non-default values for IPsec and IKE tunnel options, or to enable options that are disabled by default options, such as Network Address Translation Traversal (NAT-T) or an IKEv2 authentication method. Unless you need to configure one or more non-default options, you do not have to complete the remaining steps.

11. (Optional) Enter the following **ikev2** commands at the global configuration level to enable the NAT-T option, and to specify the NAT keep-alive time interval.

    In this example, the option is enabled and the keep-alive time interval is set to 30 seconds (the default is 20 seconds).

```
device(config)# ikev2 nat-enable
device(config)# ikev2 nat keepalive 30
```

12. (Optional) Enter the **ikev2 proposal** command to define a non-default proposal for the initial phase of the IKEv2 peer negotiation, and to enter IKEv2 proposal configuration mode. (You must be in IKEv2 proposal configuration mode configure a non-default proposal.)
    In this example, an IKEv2 proposal named *proposal1* is defined.

```
device(config)# ikev2 proposal proposal 1
device(config-ikev2-proposal)#
```

13. (Optional) Enter the **config-ike-proposal** command to select a non-default Diffie Hellman (DH) group, or groups. (The default DH group is 20. The non-default groups you can select are groups 14 or 19.)
    In this example, DH group 14 is selected and the default (DH group 20) is disabled. Only DH group 14 will be included in the IKEv2 proposal. The default is disabled to ensure it is not selected during the negotiation, because if multiple DH groups are selected, the first matching DH group supported by both ends is automatically selected.)

```
device(config-ikev2-proposal-proposal 1)# dhgroup 14
device(config-ikev2-proposal-proposal 1)# no dhgroup 20
```

14. Enter the **ikev2 policy** command at the IKEv2 configuration level to bind to protect IKE during negotiations and enter IKEv2 policy configuration mode.

```
device(config-ikev2)# ikev2 policy test-policy
device(config-ikev2-policy)# proposal 1
```

15. Enter the **ikev2 profile** command at the IKEv2 configuration level to add an authentication profile applied for the incoming IKE sessions and enter IKEv2 profile configuration mode.

```
device(config-ikev2)# ikev2 profile test-profile
device(config-ikev2-profile)#
```

NOTE

If you selected PKI-based or pre-shared key authentication, it is recommended that you select the following IKEv2 profile options:

- **PKI-based authentication**: Select Distinguished Name (DN).
- **Pre-shared key authentication**: Select Fully Qualified Domain Name (FQDN).

Step **16** is only required if you generated key pairs using ECDSA by completing the first 3 steps of this procedure. If you did not generated key pairs using ECDSA, go directly to step **16**.

NOTE
Pre-shared key is the default, but you can change the value or format of the pre- shared key. You also have the option of configuring the pre-shared key in either text or hex format.

16. (Optional) Enter one of the following commands to select the IKEv2 authentication method. You can choose ECDSA P-384 or P-256.

```
device(config-ikev2)# method <local | remote> ecdsa384
device(config-ikev2)# method <local | remote> ecdsa256
```

17. Enter the **ikev2 auth-proposal** command at the IKEv2 configuration level for IKE peer authentication and enter IKEv2 authentication configuration mode.

```
device(config-ikev2)# ikev2 auth-proposal test-authentication
device(config-ikev2-auth)#
```

18. Enter the **ipsec proposal** command at the IPsec configuration level to specify the IPsec encryption parameters used in the IPsec policy and enter IPsec proposal configuration mode.

```
device(config-ipsec)# ipsec proposal test-proposal
device(config-ipsec-proposal)#
```

ACL for a port within the IPsec tunnel

The following lists Access Control Lists (ACLs) for a port to be used within the IPsec tunnel:

- access-list 118 sequence 5 permit udp any host 10.20.81.105 eq isakmp log
- access-list 118 sequence 7 permit udp any host 10.20.81.105 eq ntp log
- access-list 118 sequence 9 permit udp any host 192.168.96.2 log
- access-list 118 sequence 10 permit tcp any host 192.168.96.2 log
- access-list 118 sequence 12 permit icmp any host 192.168.96.2 any-icmp-type log
- access-list 118 sequence 20 deny ip any any log

You configure ACLs on a global basis, then apply them to the incoming or outgoing traffic on specific ports. You can apply only one IPv4 ACL to a port's inbound traffic and similarly, only one IPv4 ACL to a port's outbound traffic. The software applies the entries within an ACL in the order they appear in the ACL's configuration. As soon as a match is found, the software takes the action specified in the ACL entry (permit or deny the packet) and stops further comparison for that packet.

NOTE
In order for the TOE to be configured to meet the NDPP with VPN gateway requirements, the last rule configured for every interface must be the one which denies all traffic that is not already explicitly permitted by a previous rule.

SPD rules

Each SPD rule requires two ACL rules, one defining the traffic flows on the tunnel, and one defining the traffic flows within the tunnel. The protocol to which the SPD rule applies must be identical for both ACL rules, and the sequence number used with the ACL rules, should be consecutive.

SPD rule	Action	Address	Protocol	Sequence number
BYPASS	Permit	Public address	Applicable protocol	N

SPD rule	Action	Address	Protocol	Sequence number
	Deny	Tunnel internal address	Applicable protocol	N + 1
PROTECT	Deny	Public address	Applicable protocol	N
	Permit	Tunnel internal address	Applicable protocol	N + 1
DISCARD	Deny	Public address	Applicable protocol	N
	Deny	Tunnel internal address	Applicable protocol	N + 1

Entering Common Criteria Operational mode

When the device is in Common Criteria Administrative mode, perform the following steps to place the device into Common Criteria Operational mode.

1. Configure the local user accounts as secure and delete non-secure user accounts. A local user account is secure when it has a password with characters from three or more character classes. These character classes are uppercase, lowercase, numeric, and ASCII non-alphanumeric characters.
2. Configure secure logging by setting up the encrypted syslog server. For details, refer to Configure an encrypted syslog server.
3. Use the **enable aaa console** command to ensure user authentication during the next reload. This also requires that you have enabled AAA authentication with the **aaa authentication login default** command.
4. Use the **logging cli-command** command to allow you to log all syntactically valid CLI commands from each user session into the system log.
5. Use the **write memory** command to save the configuration.
6. Reload the device.

On successful completion of these steps, the device will be in Common Criteria Operational mode.

Displaying Common Criteria information

After you have enabled Common Criteria Administrative mode on the device, you can display the information with the **fips show** command.

```
device# fips show
FIPS Validated Cryptographic Module
FIPS Version: BRCD-IP-CRYPTO-VER-3.0a
FIPS mode   : Administrative status ON: Operational status ON
FIPS CC mode: Administrative status ON: Operational status ON

System Specific:
OS monitor access status is: Enabled

Management Protocol Specific:
Telnet server                 : Disabled
Telnet client                 : Disabled
TFTP client                   : Enabled
SNMP v1, v2, v2c              : Disabled
SNMP Access to security objects: Enabled
Password Display              : Disabled
Any AAA server (including     :
                TACACS, None) : 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                       : Retain
```

NOTE

The HTTPS RSA host keys and signature are for the MLXe chassis only; not available for the NetIron CER device.

After you have enabled Common Criteria Operational mode by zeroizing the FIPS keys, saving the configuration, and reloading the device, enter the **fips show** command to verify the operational mode status.

```
device# fips show
FIPS Validated Cryptographic Module
MP FIPS Version: BRCD-IP-CRYPTO-VER-3.0
LP FIPS Version: BRCD-LP-CRYPTO-VER-1.0
FIPS mode : Administrative status ON: Operational status ON
FIPS CC mode: Administrative status ON: Operational status ON
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
Any AAA server (including               :
RADIUS, non TLS-TACACS+, None)          : 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
```

TABLE 11 fips show command output description

Field	Description
OS monitor access status is	The following policy allows full access to the OS monitor mode. This includes read, write access for debug purposes: **fips policy allow monitor-full-access**.
Telnet server	Telnet client and server are always disabled in FIPS CC Operational mode.
Telnet client	Telnet client and server are always disabled in FIPS CC Operational mode.
TFTP client	To allow TFTP access in FIPS mode, use **fips policy allow tftp-access**.
HTTPS SSL 3.0	Always disabled in FIPS mode.
SNMP v1, v2, v2c	Always disabled in FIPS CC mode. SNMPv3 in noAuthNoPriv, and authNoPriv security mode is not supported. Only SNMPv3 in authPriv security mode is supported.
SNMP Access to security objects	To allow SNMP read access to the critical security parameters and MIB objects, use **fips allow snmp-csp-access**.
Password Display	To allow password display, use **fips allow password-display**.
Any AAA server	To allow any AAA server (including RADIUS and non-TLS-Encrypted TACACS+ servers) to be used in FIPS CC mode, use **fips policy allow common-criteria aaa-server-any**.
Protocol shared secret and host passwords	To retain the protocol shared secrets and host access passwords between FIPS mode and non-FIPS mode, use **fips policy retain shared-secrets**.
HTTPS DSA Host keys	To retain the SSH RSA host keys between FIPS mode and non-FIPS mode, use fips policy retain rsa-host-keys (for MLX platform only).

NOTE

Making changes to the default FIPS security policy weakens the security of the device and makes the device non-compliant with FIPS 140 Level 2. 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, Brocade does not recommend making changes to the default security policy at any time.

Encrypted syslog servers in Common Criteria mode

NetIron devices in any mode send the generated syslog messages in real time to the local log storage on the device and to a syslog server (only if a syslog server is configured and available).

NOTE
Starting with Brocade NetIron R05.9.00a, using the encrypted syslog server in CC mode is optional.

A NetIron device running in Common Criteria operational mode queues the syslog messages if a syslog server is not available or configured for the device. This queue is not related to the local syslog messages store and it is cleared when the syslog messages in the queue are forwarded to the syslog server. The queue cannot hold more than 3,000 syslog messages. On reaching the maximum message limit, the device displays an error message and no further syslog messages are queued.

NetIron devices, when enabled for Common Criteria mode, do not support syslog servers that use UDP transport. However, other parameters that are defined for syslog server connections, such as specifying the hold time for queued messages and traps when the device reloads or switches over are applicable for encrypted syslog connections as well.

When you enable Common Criteria mode on a device, the device is in the Common Criteria Administrative mode, where syslog server configuration that uses UDP transport is retained. You can configure encrypted syslog server connections in this mode. However, syslog messages that are generated when the device is in the administrative mode are sent to the UDP syslog servers, not to the encrypted syslog server that you have configured. When the device is put in the Common Criteria Operational mode, existing syslog servers that use UDP transport are removed, and only encrypted syslog server connections are accepted.

Conversely, when a device is downgraded from Common Criteria mode, the encrypted syslog server connections that were configured are removed, and the device supports only unencrypted UDP syslog servers. The following table summarizes these transitions.

TABLE 12 Syslog server connections during transition to and from Common Criteria mode

From	To non-FIPS mode	To FIPS mode	To Common Criteria Operational mode
Non-FIPS mode	Not applicable	No change. FIPS mode does not support encrypted syslog servers.	All the UDP servers are removed when the device is put in CC Operational mode. Only encrypted syslog server connections are allowed in CC Operational mode.
FIPS mode	No change	Not applicable	All the UDP servers are removed when the device is put in CC Operational mode. Only encrypted syslog server connections are allowed in CC Operational mode.
Common Criteria mode	All the SSL servers are removed. Non-FIPS mode does not support encrypted syslog server connections.	Not allowed. You must disable Common Criteria mode to revert to non-FIPS mode, and then re-enable FIPS mode. FIPS mode does not support encrypted syslog server connections.	Not applicable

AAA servers in Common Criteria mode

Common Criteria mode requires that devices support NDPP version 1.1. This protocol defines the communication of the device with AAA servers to take place over a TLS-encrypted session.

Even though you can configure multiple TLS-encrypted TACACS+ servers, only one connection can be active at any time. If another TLS-encrypted TACACS+ session is attempted at the same time as the first TACACS+ session, the connection attempt is rejected.

Additionally, since the TACACS+ server may accept only a single TACACS+ session over the TCP or the TLS-encrypted connection, it is recommended you use this only for authentication.

When the device is in Common Criteria Operational mode, and the device has been configured for a TLS encrypted TACACS+ server for authentication, only one administrator will be able to administer the device. In addition, accounting and authorizing using the TLS-encrypted TACACS+ server will be disabled.

> NOTE
> You can modify the default Common Criteria policy to allow a non-TLS-encrypted TACACS+ server, but this will make the device noncompliant with Common Criteria requirements.

Modifying the Common Criteria policies to use non-encrypted AAA servers

If required, you can modify the Common Criteria policies to allow AAA servers that do not use TLS encryption to be configured, such as RADIUS servers. When non-encrypted AAA servers are allowed, you cannot configure TLS-encrypted TACACS+ servers on the device.

> NOTE
> Modifying the default Common Criteria policy will make the device noncompliant with Common Criteria standards.

To allow any AAA server to work with the device in Common Criteria mode, enter the following command:

```
device# fips policy allow common-criteria aaa-server-any
```

Syntax: [no] fips policy allow common-criteria aaa-server-any

Use the **no** form of the command to remove non-encrypted AAA servers. If any non-encrypted AAA servers are available on the device, they are removed when Common Criteria mode is enabled on the device.

Use the **show aaa** command for TLS-encrypted TACACS+ servers.

```
device # show aaa
TACACS default key: ...
TACACS retries: 1
TACACS timeout: 5 seconds
TACACS+ Server: IP=10.25.105.201 SSL-Auth-Port=60520 Usage=any Key=...
opens=0 closes=0 timeouts=0 errors=0
packets in=0 packets out=0
no connection
***** Radius server not configured
```

Downgrading from Common Criteria mode to non-FIPS mode

Downgrading a device from Common Criteria mode to either FIPS mode or non-FIPS mode uses the same command. You cannot directly downgrade to FIPS mode; you first downgrade to non-FIPS mode, and then enable FIPS mode using the procedures detailed in the earlier chapter.

After the device is placed in non-FIPS mode, you can use SCP to download and initialize an older image. Use the following steps to revert to a non-FIPS-compliant image.

1. Log in to the device by entering your username and password.
2. Disable Common Criteria mode by entering the **no fips enable** or **no fips enable common-criteria** command.
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.
5. Reload the configuration by entering the **reload** command.

Commercial Solutions for Classified program

The Commercial Solutions for Classified (CSfC) program was established by the United States government to enable commercial networking applications to be used in layered solutions protecting classified National Security Systems (NSS) data. The CSfC program provides the ability to securely communicate based on commercial standards in a solution. Brocade, as a networking company, supports CSfC and many of the products listed in the CSfC component list. Brocade devices must be approved by the CSfC program to be deployed in government or federal networks.

Network Device Protection Profile with VPN gateway

The Network Device Protection Profile (NDPP) standards provide a set of rules that define the security requirements for network devices. The main purpose of these requirements is to minimize and reduce threats to network devices. NDPP requires SSH or TLS for syslog and authentication server communications while NDPP with VPN gateway requires IPsec for syslog and authentication server communications.

The BR-MLX-10Gx4-IPSEC-M interface module supports creation of a VPN using the IPsec protocol and also the setup of security associations (SAs) for the IPsec protocol suite. Once the SAs are set up, the IPsec protocol is used to set up and operate encrypted tunnels between two endpoints. NDPP with VPN gateway allows the Brocade NetIron MLXe to be used as a VPN gateway within highly secure and high security federal networks. These networks should be approved by the Commercial Solutions for Classified (CSfC) program.

NOTE
The connections syslog and AAA servers need to be over TLS only. For more information see the output of **fips enable common-criteria** command with lines prefixed with "CC".

NDPP with VPN gateway requirements

- The management module traffic, including SSH and HTTPS, should be transported over the IPsec network provided by the BR-MLX-10Gx4-IPSEC-M interface module.

  NOTE
  The management module does not have any support for an IPsec stack.

- Elliptic curve-based key establishment support curves.
- The TSF should ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP), and 20 (384-bit Random ECP).
- New audit events for IKE and IPsec.
- Support for IKEv2 as defined in RFC 5996.
- Support for AES-256-CBC and AES-128-CBC for IKEv2.
- Support for AES256-GCM and AES-128-GCM for IPsec.
- Support for binary bits-based PSK for IKEv2 authentication.
- Support for X509v3 certificate for authentication of IKEv2 endpoints using ECDSA P-384 and P-256 curves.

- Support for NAT traversal.
- TOE supports IPv4 per RFC 791, IPv6 per RFC 2460, TCP per RFC 793, and UDP per RFC 768.

NOTE
If the **enable strict-password-enforcement** command is enabled, users have up to three login attempts. If a user fails to login after third attempts, that user is locked out (disabled). Enable the user by entering the **username name enable** command.

NAT traversal in IKE and IPsec

The VPN gateway Extended Profile (EP) requires that IKEv2 and IPsec tunnel to support Network Address Translation (NAT) traversal for a client. The peer device is set up behind a NAT device or a NAT firewall to protect its identity.

IPsec protocol protects the integrity of the IP packet besides encapsulation and encryption. When an IP packet passes through a NAT device, there is a change in the IP/TCP header leading to the violation of integrity and thus the packet is discarded by the IPsec tunnel end nodes. To overcome this issue, determine whether the IKE peers are capable of supporting NAT traversal, or, or if there is a NAT device between the IKE peers.

For more information about NAT traversal, refer to the specific sections in the *Brocade NetIron Security Configuration Guide.*

The following commands are introduced as part of NAT traversal in IKE and IPsec:

- **ikev2 nat-enable**
- **ikev2 nat-keepalive** *time in seconds*

NOTE
IPsec supports only the IKEv2 protocol; the IKEv1 protocol is not supported.

Selecting the AES-GCM-128 algorithm

You select the encryption algorithm for the tunnel when configuring the IPsec proposal.

The following example selects the AES-GCM-128 algorithm for the IPsec proposal named *ipsec_proposal*.

```
device(config)#ipsec proposal ipsec_proposal
device(config-ipsec-proposal-ipsec_proposal)#encryption-algorithm aes-gcm-128
```

The following example displays the configuration of an IPsec proposal named *ipsec_proposal*. AES-GCM-128 is the configured encryption algorithm.

```
device#show ipsec proposal ipsec_proposal
================================================================================
Name                : ipsec_proposal
Protocol            : ESP
Encryption          : aes-gcm-256,aes-gcm-128
Authentication      : NULL
ESN                 : Disable
Mode                : Tunnel
Ref Count           : 0
```

Audit logging

Brocade NetIron provides support for logging of IKE and PKI transaction details. The logs are automatically generated syslog messages that contain the IKEv2 and PKI transaction details.

There are two types or levels of logging. Standard (default) logging is enabled by default. The second type of logging is called extended logging, which you must enable using commands. This type of logging allows you to log additional IKE or PKI transaction details.

Configuring the strict password rules

Use the **enablestrict-password-enforcement** command to enable the strict password enforcement feature. Enter a command such as the following.

```
device(config)# enable strict-password-enforcement
```

Syntax: [no] enable strict-password-enforcement

This feature is disabled by default.

When enabled, the system verifies uniqueness against the history of passwords of the user whose password is being set. Passwords must not share four or more concurrent characters with any other password configured for that user on the device. If the user tries to create a password which shares four or more concurrent characters for that user, the following error message is returned:

```
Error - The substring <str> within the password has been used earlier, please choose a different password.
```

Also, if the user tries to configure a password that was previously configured, the local user account configuration is not allowed and the following message is displayed.

```
Error - This password was used earlier, please choose a different password.
```

When you create a password, the characters you type are masked.

: To assign a password for a user account.

```
device(config)# username sandy password [Enter]
Enter new password: ********
```

Syntax: [no] username name password

Enter a password such as TesT12$! that contains the required character combination.

> NOTE
> If enable strict-password-enforcement is enabled, when a user is logged in and is attempting to change their own user password, the following prompt is displayed: Enter old password. After validating the old password, the following prompt is displayed: Enter new password.

Support for Logging IKE and PKI Transaction Details

Brocade NetIron provides support for logging of IKE and PKI transaction details. The log files are automatically generated syslog messages that contain the transaction details.

There are two types or levels of logging. Standard (default) logging is enabled by default. The second type of logging is called extended logging, which you must enable using commands. This type of logging allows you to log additional IKE or PKI transaction details.

Required hardware

The hardware requirements are identical for default logging and extended logging. The following table lists the required hardware.

Required Hardware	Required

	Hardware
Device	Brocade MLXe router
Line card	Brocade IPsec module (BR-MLX-10GX4-IPSEC

	- M)

Limitations

All of the current limitations of the logging feature on MLX devices, and the limitations of the IPsec security feature apply to the logging of IKE and PKI transaction details.

In addition, there are some limitations specific to the feature for logging IKE and PKI transaction details. The following table lists the current limitations for this feature.

Default and Extended Logging	**D e s c r i p t i o n**
IKE transaction details (send and receive packets)	T h e m a x i m u m s i z e s y s l o

g b u ff e r i s 1 0 2 4 b y t e s . L o g g i n g o f p a c k e t c o n t e n t i n h e x f o r m a t a l o n g w i t h o t

	herloggingparametersinthepacketsyslogonlyallow250bytesofpacket

in one syslog. An IKEv2 packet that attempt to get her with protocol header stotals

	more than 250 bytes will be logged in multiple syslogs. If all in each card down which
Packet and event syslogs for IKE sessions	

chIPsectunnelsareconfiguredd(IPv4orIPv6tunnels)isrebootted,packet

and event data for IKE sessions are lost. The packet and event data are not l

ogged on the local syslog or the remote syslog server.

Management commands

The following list of commands and command variants are required for administration of the TOE. These commands are available only after an administrator has successfully logged into the TOE.

TABLE 13 Management commands

Command	Tested Command Variants	Description
write	**write memory**	Write to persistent storage
crypto	**crypto key generate**	Invoke cryptographic functions
openssl	**openssl s_server**	Configure secure connections (for example with syslog)
logging	**logging host ssl-port***ip-address* ssl-port *port*	Configure the audit logging host

TABLE 13 Management commands (continued)

Command	Tested Command Variants	Description
reload	**reload**	Reload the current flash image
console	**console timeout** *time*	Manage console properties
banner	**banner** motd+	Manage the login banner
exit	**exit**	Log out or exit current session
ntp	**ntp**	Switch to NTP configuration mode
config	**config** terminal	Switch to configuration mode
username	**username** *user* password	Manage user accounts
clock	**clock** *set time*	Manage the internal clock
server	**server** *ntp server ip*minpoll *time*	Configures external services
crypto-ssl	**crypto-ssl** certificate generate	Manages web server properties
web-management	**web-management** session-timeout *time*	Manages web interface
fips	**fips** enable common-criteria **fips** show **fips** zeroize all	Manages FIPS and common criteria configuration
aaa	**aaa** authentication **aaa** authentication enable default tacacs+ local **aaa** authentication login default tacacs+ local **aaa** authentication web-server default local	Configures the AAA authentication functions
radius-server	**radius-server** host *ip address* ssl-auth-port *port* **radius-server** retransmit *retransmit period* **radius-server** timeout *timeout period* **radius-server** key *key name*	Configures the RADIUS server
tacacs-server	**tacacs-server** host *ip address* ssl-auth-port *port* default **tacacs-server** retransmit *retransmit period* **tacacs-server** timeout *timeout period* **tacacs-server** key *key name*	Configures the TACACS+ server
ikev2	**ikev2** proposal **ike** profile	Configures the IKEv2 properties
ipsec	**ipsec** proposal **ipsec** profile	Configures IPsec properties
tunnel	**tunnel** protection ipsec ipv4 *ipsec profile name*	Enables IPsec on an interface
ip / ipv6	**access-list**	Configures IPv4 and IPv6 ACLs
enable	**enable** aaa **enable** password-min-length 15	Enables console login features
show	**show** flash **show** version **show** clock **show** ip client-pub-key **show** ip ssl **show** logging **show** run	Displays information about specified configuration
access-list	**access-list**deny host *IP address*	Creates ACL rules

TABLE 13 Management commands (continued)

Command	Tested Command Variants	Description
	access-list 1 deny 10.157.29.12 **access-list** 1 deny host IPHost1 **access-list** 1 permit any	
interface	**interface** ethernet 4/12 **interface** mac access-group 400 in	Associates an ACL with an interface
lifetime	**lifetime** *lifetime in minutes*	Configures the IKEv2 security association (SA) lifetime value in minutes in the IKEv2 profile config mode. For example: `device(config-ike-profile-ipsec-linux)# lifetime 400`
lifetime	**lifetime** *lifetime in minutes*	Configures the IPsec SA lifetime value in minutes in the IPsec profile config mode. For example: `device(config-ipsec-profile-ipsec-linux)# lifetime 600`

Configuring an Encrypted Syslog Server

- Encrypted syslog server overview..... 85
- TLS encrypted syslog server configuration and validation..... 87

Encrypted syslog server overview

The information available in this appendix is a representative configuration example of the many types of syslog servers available.

Though there are many types of syslog servers available, the following setup procedure describes how to set up an encrypted syslog server running on Ubuntu 10.4. The setup procedure for an encrypted syslog server on other Linux operating systems such as Red Hat or Centos is simiilar except for the differences in commands.

You must set up stunnel as a server and a client on your server. As a server, stunnel listens on port 60516 to connections from its client peers, and all connections are forwarded to the locally-running rsyslog listening at port 61514. As a client, rsyslog forwards messages to the stunnel local portal at port 61514, and stunnel local port forwards data by way of the network to port 60514 to its remote peer.

Setting up stunnel

1. Install the stunnel utility with the following command:

```
$ sudo apt-get install stunnel4
```

2. Edit the file with the `/etc/default/stunnel4` path to start the service on system startup. Use a text editor such as vi.

```
$ sudo vi /etc/default/stunnel4
```

3. Change the line `Enabled=0` to `Enabled=1`.

Creating a certificate with the OpenSSL toolkit

1. Enter the following command:

```
cd /etc/stunnel
```

2. To create the `/etc/stunnel/stunnel.pem` file with a certificate and key for SSL, enter the following command

```
$openssl req -new -x509 -days 365 -nodes -out stunnel.pem -keyout
/etc/stunnel/stunnel.pem
```

3. To change the permissions for the certificate that you generated, enter the following command.

```
$ sudo chmod 600 /etc/stunnel/stunnel.pem
```

Creating a configuration file

1. Enter the following command to open the stunnel.conf file:

```
$sudo vi /etc/stunnel/stunnel.conf
```

2. Comment out the features that you do not require, such as the [pop3s], [ssmtp], and [imaps] sections.

3. Change the `cert=/etc/stunnel/mail.pem` line to `cert=/etc/stunnel/stunnel.pem`.
4. Add the following lines and save the file.

```
; Certificate/key is needed in server mode
cert = /etc/stunnel/stunnel.pem
key = /etc/stunnel/stunnel.pem

; Some debugging stuff useful for troubleshooting
debug = 7
foreground=yes

[ssyslog]
accept = 60514
connect = 61514
```

Changing the stunnel4 startup file

Enter the `cd /etc/init.d/stunnel4` command and change `ENABLED=0` to `ENABLED=1`.

Restarting the stunnel service

To restart the stunnel service, enter the following command.

```
$sudo /etc/init.d/stunnel4 restart
```

Configuring rsyslog

Ubuntu 10.04.3 comes with rsyslog 4.2.0 as its default logger. You can add MySQL output support and the Reliable Event Logging Protocol (RELP). Enter the following command:

```
root@linux:~$sudo apt-get install rsyslog-mysql rsyslog-relp
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
        dbconfig-common librelp0
The following NEW packages will be installed:
        dbconfig-common librelp0 rsyslog-mysql rsyslog-relp
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 677kB of archives.
After this operation, 2,335kB of additional disk space will be used.
Do you want to continue [Y/n]? y
```

During the installation process, complete the following steps:

1. Create the tables that are needed in MySQL when prompted.
2. Set the MySQL root password.
3. Create a password that the rsyslog processes will use in the configuration files.

Enabling accepting remote logs

To turn on accepting remote logs, edit the `/etc/rsyslog.conf` file by commenting out the following lines:

```
# provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 61514
```

Restarting rsyslog service

To restart the rsyslog service, enter the following command:

```
root@linux:~$sudo service rsyslog restart
```

NOTE
Brocade recommends rebooting the Linux server after the setup.

Printing log messages

Enter the following command to update the log-watcher window with logged messages as they arrive:

```
root@linux:~$tail -f /var/log/messages
```

You can also configure a web user interface to display the syslog messages using the Reliable Event Logging Protocol (RELP). Refer to http://www.linuxjournal.com/content/centralized-logging-web-interface for more information.

TLS encrypted syslog server configuration and validation

Certificates (both server and trusted) must meet the following criteria.

- Only RSA certificates are accepted.
- The public key must be greater than or equal to 2048 bits.
- The Signature Algorithm must be using SHA256.
- The device must have a server certificate installed.
- An expired certificate is not accepted.
- A certificate with an empty Subject Alternative Name (SAN) field is rejected.
- When the server's certificate signature is invalid, the client rejects a certificate based on the public key provided in the issuer's self-signed certificate.
- A certificate with a mismatching Subject Alternative Name (SAN) IP address field is rejected.
- Certificates must use the correct cipher suites.

In Common Criteria mode, when the device acts as a TLS client while connecting to a remote server, the client must perform validation of the server certificate.

1. Create the TLS encrypted syslog server's private key using the **openssl genrsa** command.

```
        openssl genrsa -out rsakey2048.pem 2048
        Generating RSA private key, 2048 bit long modulus
        ..+++
.....................................................+++
e is 65537 (0x10001)
```

2. Create the TLS encrypted syslog server's self-signed certificate, also including the IP address of the server in the Subject Alternative Name (SAN) field of the certificate.

   a) Create a configuration file that looks like the following example.

```
cat req_san.config.txt

[ req ]
default_bits            = 2048         # Size of keys
default_keyfile         = key.pem     # name of generated keys
default_md              = sha256     # message
digest algorithm
string_mask             = nombstr     # permitted characters
distinguished_name      = req_distinguished_name
req_extensions          = v3_req

[ req_distinguished_name ]
# Variable name Prompt string
#------------------------ ----------------------------------
0.organizationName      = Organization Name (company)
organizationalUnitName  = Organizational Unit Name (department,
division)
emailAddress            = Email Address
emailAddress_max        = 40
localityName            = Locality Name (city, district)
stateOrProvinceName     = State or Province Name (full name)
countryName             = Country Name (2 letter code)
countryName_min         = 2
countryName_max         = 2
commonName              = Common Name (hostname, IP, or your name)
commonName_max          = 64

[ v3_req ]
basicConstraints        = CA:FALSE
subjectKeyIdentifier    = hash

[ extensions_section ]
subjectAltName          =IP:192.168.10.201
```

   b) Create the certificate, giving the configuration file created in the previous step as a parameter.

```
openssl req -new -x509 -key rsakey2048.pem -out
rsacert2048_days1095_sha256_SAN.pem -days 1095 -sha256 -config
./req_san.config.txt -extensions extensions_section

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Organization Name (company) []:Brocade
Organizational Unit Name (department, division) []:Engineering
Email Address []:
Locality Name (city, district) []:San Jose
State or Province Name (full name) []:California
Country Name (2 letter code) []:US
Common Name (hostname, IP, or your name) []:SP_EMIS TLS Encrypted SYSLOG
```

   c) To view the TLS encrypted syslog server's self-signed certificate that is created, enter the command as shown.

```
openssl x509 -inform PEM -noout -text -in rsacert2048_days1095_sha256_SAN.pem

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            f9:aa:bd:da:1b:5a:3e:51
        Signature Algorithm: sha256WithRSAEncryption
```

```
        Issuer: O=XYZ, OU=ABC, L=San Jose, ST=California, C=US, CN=SP_EMIS TLS Encrypted SYSLOG
server
        Validity
            Not Before: Mar 31 22:20:47 2014 GMT
            Not After : Mar 30 22:20:47 2017 GMT
        Subject: O=XYZ, OU=ABC, L=San Jose, ST=California, C=US, CN=SP_EMIS TLS Encrypted SYSLOG
server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ca:5f:78:de:07:b7:15:21:b4:9d:e9:66:b7:5e:
                    48:8b:96:ed:4b:f3:5d:dc:d7:95:27:ed:ca:1d:00:
                    9d:d6:06:5b:f5:df:d2:0c:54:69:53:4a:38:d1:52:
                    2d:bf:6c:a4:2b:7d:dd:ad:e7:2c:5a:4f:1c:0e:8b:
                    59:7a:04:f1:54:b8:00:99:51:21:f7:42:81:17:4c:
                    cc:94:86:00:8b:c6:c0:0d:3b:7a:19:66:3c:e5:33:
                    be:5f:b5:2c:d9:df:74:1c:07:f5:41:82:c0:b2:48:
                    9e:c3:7b:cc:2e:07:4e:d8:2a:17:69:48:ae:f2:97:
                    4a:fd:7e:4b:34:2d:36:49:bb:3a:79:c6:c4:9c:1e:
                    5f:1b:d7:59:a0:3e:27:02:2f:2b:eb:60:26:95:20:
                    bb:2a:e8:5b:9b:56:b6:2e:62:eb:a1:21:f4:95:1c:
                    e1:d6:ca:4e:74:0a:a1:6a:f6:b0:27:7f:f4:e2:d2:
                    92:f9:db:25:49:9f:c1:87:d3:ed:1f:d1:98:6c:da:
                    15:04:c1:bb:16:66:78:02:ab:81:a0:98:c2:62:75:
                    b1:4e:96:0a:fd:25:84:64:f3:e6:35:5e:06:05:79:
                    c6:83:73:d6:33:6b:57:64:ad:4d:b5:f4:3d:f6:e7:
                    e5:a3:71:d0:c9:e5:77:7a:4a:11:c0:89:ca:1a:35:
                    72:df
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                IP Address:192.168.10.201
    Signature Algorithm: sha256WithRSAEncryption
        88:a5:6c:d3:15:c2:10:20:c9:36:73:ba:c5:72:4c:e4:26:78:
        dc:ec:21:a2:2b:ec:4b:5a:42:85:be:fe:c4:1f:01:97:f0:5c:
        e2:51:1a:3b:84:15:c9:cb:63:35:b1:e6:b8:3e:2a:76:47:5f:
        ce:1b:59:80:43:81:95:b8:aa:1b:11:7f:80:6f:3f:97:d9:0c:
        43:7e:53:b0:04:80:be:52:da:4b:0b:b4:70:07:a8:b6:d8:09:
        55:9f:4e:08:7a:c1:df:7e:da:dd:c0:59:f3:9d:c6:f5:2b:ec:
        66:89:9e:c9:5f:6c:d1:e7:fe:1e:d3:b1:6e:9f:84:3c:fb:ed:
        e5:c9:2c:7f:8c:85:f4:97:bb:99:3c:cd:1e:3e:d2:a1:6e:09:
        3a:05:b6:c1:76:b9:54:ec:34:a8:a9:6f:ca:30:34:cb:ec:05:
        5d:17:a3:cb:21:3a:69:e3:7d:28:d2:15:c0:19:0e:34:00:8d:
        68:ce:cd:0a:65:db:e4:88:b6:d1:67:40:3a:3d:22:bf:dc:22:
        16:ec:4f:08:a7:54:7f:42:73:9b:f7:88:1a:70:73:8c:81:a8:
        5c:b4:55:5f:7e:94:75:ec:93:f0:48:08:18:4a:3c:ea:c6:6b:
        48:d6:b2:f4:ff:de:23:df:d5:fd:a0:bd:8a:cb:c7:69:f9:3a:
        d9:5f:c5:0f
```

3. Upload the self-signed certificate to the device. Verify the certificate after upload, using either the signature or the fingerprint of the certificate.

```
scp rsacert2048_256_days1095_SAN.pem lab1@192.168.105.82:ssltrustedcert

lab1@192.168.105.82's password:
rsacert2048_256_days1095_SAN.pem                100% 1448                1.4KB/s
00:00
Connection to 192.168.105.82 closed by remote host.
```

   The **scp** command can be executed from a remote system (such as Linux or Windows). Use the **ssltrustedcert** option to upload a trusted certificate (certificate.pem) to the device specified by the IP address as user.

   The device can have up to three dynamic trusted certificates. Once three dynamic trusted certificates are uploaded, running the command again returns an error.

   To display the list of the dynamic trusted certificates on the device, use the **show ip ssl certificate** command.

   To delete the dynamic trusted certificate list on the device, use an empty certificate file. Following example deletes the dynamic trusted certificate list:

```
> ls -la empty.file
ls: empty.file: No such file or directory
> empty.file
> ls -la empty.file
-rw-r--r-- 1 lab engr 0 Mar 31 12:47 empty.file
> scp empty.file lab@192.168.10.82:ssltrustedcert
```

4. Verify the certificate after upload to the device, using either the signature or the fingerprint of the certificate.

   The **show ip ssl certificate** command displays the dynamic trusted certificate list. The dynamic trusted certificate list can be modified by the **scp ssltrustedcert** command.

```
device# show ip ssl certificate

No SSL sessions in use.
Trusted Certificates:
    Dynamic:
        Signature Algorithm: sha256WithRSAEncryption
        Validity:
            Not Before: Mar 31 2014 13:22:47
            Not After : Mar 30 2017 13:22:47
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                IP Address:192.168.10.201
        Signature:
            88:a5:6c:d3:15:c2:10:20:c9:36:73:ba:c5:72:4c:e4:26:78:
            dc:ec:21:a2:2b:ec:4b:5a:42:85:be:fe:c4:1f:01:97:f0:5c:
            e2:51:1a:3b:84:15:c9:cb:63:35:b1:e6:b8:3e:2a:76:47:5f:
            ce:1b:59:80:43:81:95:b8:aa:1b:11:7f:80:6f:3f:97:d9:0c:
            43:7e:53:b0:04:80:be:52:da:4b:0b:b4:70:07:a8:b6:d8:09:
            55:9f:4e:08:7a:c1:df:7e:da:dd:c0:59:f3:9d:c6:f5:2b:ec:
            66:89:9e:c9:5f:6c:d1:e7:fe:1e:d3:b1:6e:9f:84:3c:fb:ed:
            e5:c9:2c:7f:8c:85:f4:97:bb:99:3c:cd:1e:3e:d2:a1:6e:09:
            3a:05:b6:c1:76:b9:54:ec:34:a8:a9:6f:ca:30:34:cb:ec:05:
            5d:17:a3:cb:21:3a:69:e3:7d:28:d2:15:c0:19:0e:34:00:8d:
            68:ce:cd:0a:65:db:e4:88:b6:d1:67:40:3a:3d:22:bf:dc:22:
            16:ec:4f:08:a7:54:7f:42:73:9b:f7:88:1a:70:73:8c:81:a8:
            5c:b4:55:5f:7e:94:75:ec:93:f0:48:08:18:4a:3c:ea:c6:6b:
            48:d6:b2:f4:ff:de:23:df:d5:fd:a0:bd:8a:cb:c7:69:f9:3a:
            d9:5f:c5:0f
```

5. Configure the TLS encrypted syslog server with the server private key, and certificate. The following example shows the successful configuration of a Linux syslog server with the DHE-RSA-AES256-SHA cipher.

```
openssl s_server -accept 60892 -cert rsacert2048_256_days1095_SAN.pem -key
rsakey2048.pem -cipher DHE-RSA-AES256-SHA

Using default temp DH parameters
Using default temp ECDH parameters
ACCEPT
-----BEGIN SSL SESSION PARAMETERS-----
MHUCAQECAgMBBAIAOQQgw8qyfvnc6W0z65juN+RuUeurjFO3qVuNXTMDQPdAGdwE
MI6hWek1E/a69dWIJ6VImumyQTTuv90P+8AzwIpb2JHc3MWliE0qZJ6wsFg4jvDQ
Y6EGAgRSXsY3ogQCAgEspAYEBAEAAAA=
-----END SSL SESSION PARAMETERS-----
Shared ciphers:AES128-SHA:AES256-SHA:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA
CIPHER is DHE-RSA-AES256-SHA
Secure Renegotiation IS NOT supported
<14>Mar 31 2014 14:58:57 XM82 FIPS mode enabled by operator from console
<14>Mar 31 2014 14:58:57 XM82 CLI CMD: "fips enable common-criteria" from
console
```

6. Configure the device with the IP address of the TLS encrypted syslog server.

```
logging host 192.158.105.82 ssl-port 60892
```

NOTE

The port number should be the same as the number used in the **openssl s_server** command in step 5.

Syslog Messages

- Syslog messages in FIPS mode..... 93

Syslog messages in FIPS mode

The following table lists some of the syslog messages in FIPS mode.

TABLE 14 FIPS syslog messages

Message level	Message	Explanation
Alert	Time is updated by NTP *server ip-address* from NO_CLOCK to *<new time>* GMT+00 *<new date>*	Indicates time is updated by an NTP server.
Alert	Clock Changed from old time *<old time>* GMT +00 *<old date>* to new time *<new time>* GMT +00 *<new date>*	Indicates time is updated using the **clock set** command.
Informational	SSH login by *user* from *src* IP *ip-address*, *src* MAC *mac-address* to USER EXEC mode using RSA as Server Host Key.	Indicates entry into the "user exec" mode for all sessions for the mentioned user. Similar message is logged for "privileged exec" mode.
Informational	SSH logout by *user* from *src* IP *ip-address*, *src* MAC *mac-address* from USER EXEC mode using RSA as Server Host Key.	Indicates exit from "user exec" mode for all sessions for the mentioned user. Similar message is logged for "privileged exec" mode.
Informational	SSH session for *user* from *src* IP *ip-address*, MAC *mac-address* in PRIVILEGED EXEC mode has timed out.	Indicates SSH logout has occurred due to timeout. Similar message is logged for "user exec" mode.
Informational	SSH session closed by *user* from *src* IP *ip-address*, MAC *mac-address* in PRIVILEGED EXEC mode.	Indicates SSH logout has occurred due to termination. Similar message is logged for "user exec" mode.
Informational	SSH session killed for *user src* IP *ip-address*, MAC *mac-address* in PRIVILEGED EXEC mode.	Indicates SSH logout has occurred because the session was killed.
Informational	Super user login success in console session.	Indicates user has logged in with super user password.
Informational	Logging CLI_CMD operation enabled by *user* from console session. "logging cli-command" by *user* from console.	Indicates audit log **logging cli-command** command is enabled.
Informational	Logging CLI_CMD operation disabled by *user* from console session.	Indicates audit log **logging cli-command** command is disabled.
Informational	"reload" by un-authenticated user from console	Indicates initiation of device reload through console.
Informational	SSL Syslog server *ip-address:portnum* is now connected.	Indicates encrypted syslog server is connected in the server end.
Informational	SSL Syslog server *ip-address:portnum* is now disconnected.	Indicates encrypted syslog server is disconnected in the server end.
Informational	SSH login by *user* from *src* IP *ip-address* from *src* MAC *mac-address* to USER EXEC mode using RSA as Server Host Key. Brocade scp -t file: secondary.sig	Indicates the SCP transfer.

TABLE 14 FIPS syslog messages (continued)

Message level	Message	Explanation
	Brocade transfer to device completed SSH logout by *user* from *src* IP *ip-address* from *src* MAC *mac-address* from USER EXEC mode using RSA as Server Host Key.	
Informational	*yyyy month dd hh:mm:ss*	Indicates the timestamp format that is used in syslog messages.
Informational	Error - Incorrect username or password in console session.	Indicates incorrect username or password.
Informational	Brocade(config)#wr m Message: "write memory" by *user* from console.	Audit log will display the commands in expanded form.
Informational	console login by *user* to USER EXEC mode.	Displays all "login" events including the user and session details. Similar message is logged for "logout" events and "privileged exec" mode.
Informational	Module in slot 2 is rebooted due to FIPS HW sec engine KAT failure	The message is displayed on the MP indicating that the module in slot 2 has restarted due to KAT failure on the MP.
Informational	Module in slot 5 is rebooted due to FIPS HW macsec engine KAT failure	The message is displayed on the MP indicating that the module in slot 5 has restarted due to MACsec engine KAT failure on the MP.
Informational	Entropy generated using the HW crypto engine on slot 2 is same as the previous ... regenerating	The message indicates that entropy generated is similar to the previously generated entropy. This message is observed after the first failure.
Informational	Module 5 is reset by mgmt (reason: FIPS KAT failure)	This message is displayed on the MP for KAT failure on LP due to the LP reset issue.
Informational	`SYSLOG: <14>Nov 23 2015 14:56:15 Security: Web login by lab from src IP 10.20.81.1` `SYSLOG: <14>Nov 23 2015 14:58:01 Security: web logout by lab from src IP 10.20.81.1`	A user has logged into or logged out of the HTTPS (web) administration mode.
Informational	`SYSLOG: <14>Dec 15 18:08:13 CER63 FIPS: Image verification passed for $$$primary-temp` `SYSLOG: <14>Dec 15 18:06:21 CER63 FIPS: Image verification failed for $$$primary-temp`	A firmware image was successfully or unsuccessfully updated and verified to the device.
Informational	Logging to all supported destinations has been enabled or disabled from the CLI. "logging on -command" by *user* from console. "no logging on -command" by *user* from console.	Indicates audit logging **logging on** and **no logging on** command.
IKEv2 transaction default logging syslog messages		
Informational	`Aug 12 01:51:20:I: IKEv2: Session Established for TNL <Tunnel ID> with Src <source-address> Dest <destinationaddress>`This message includes tunnel ID and tunnel source and destination addresses.	The session is established.

TABLE 14 FIPS syslog messages (continued)

Message level	Message	Explanation
Informational	`Aug 12 01:58:25:I: IKEv2: Session Terminated for TNL <Tunnel ID> with Src <source-address> Dest <destinationaddress> SPI <SPI number>`	The session is terminated.
Informational	`Aug 12 01:55:30:I: IKEv2: Session Rekeyed for TNL <Tunnel ID> with SPI <SPI-ID> Src <source-address> Dest <destination-address>`This message includes tunnel ID, SPI, and tunnel source and destination addresses.	IKE Session Rekey
Informational	`Aug 12 01:55:30:I: IKEv2: IPSEC SA Rekeyed for TNL <Tunnel ID> with SPI <SPI-ID> Src <source-address> Dest <destination-address>` This message includes tunnel ID, SPI, and tunnel source and destination addresses.	IPSec SA Rekey
Informational	`Aug 12 01:56:40:I: IKEv2: Session Timer Expired for TNL <Tunnel ID> with Src <source-address> Dest <destinationaddress>` This message includes tunnel ID and tunnel source and destination addresses.	Timer Expiration
Informational	`Aug 12 01:51:30:I: IKEv2: Authentication Failed for TNL <Tunnel ID> with Src <source-address> Dest <destinationaddress>` This message includes tunnel ID, tunnel source, and tunnel destination.	Authentication Failure
IKEv2 transaction extended logging syslog messages		
Informational	`Aug 12 01:51:20:I: IKEv2: Pkt rcvd with Src <source-address> Dest <destination- address> SPI <SPI-ID> Exchg <X> Pkt len: XX Seq: YY Pkt content <Hex dump of the packet from L2 header>`	Packets received by IKE peer during the IKE transaction.
Informational	`Aug 12 01:51:21:I: IKEv2: Pkt sent with Src <source-address> Dest <destination- address> SPI <SPI-ID> Exchg <X> Pkt len: XX Seq: YY Pkt content <Hex dump of the packet from L2 header>`	Packets sent by IKE peer during the IKE transaction.
Informational	`Aug 12 01:51:21:I: IKEv2: Pkt sent with Src <source-address> Dest <destination-address> SPI <SPI-ID> Exchg <X> Pkt len XX Seq:YY Frag:Z Frag len: ZZ Pkt Content <Hex dump of the packet from L2 header>`	Send or receive packets that were fragmented during the IKE transaction.
PKI transaction extended logging syslog messages		
Informational	`Aug 12 10:11:12:I: PKI: connection req is sent to host:<hostname> for trust point<trustpoint_name> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>,`	All connection request events to external entities are logged, including: • Connection request events to Certificate Authority (CA)

TABLE 14 FIPS syslog messages (continued)

Message level	Message	Explanation
	`fragment:<frag_no>: frag_len:<frag_len>:<hex_dump>`	• Connection request events to external server for to download peer certificate • Connection request events to import local certificates
Informational	`Aug 12 10:11:12:I: pki: connection to host:<hostname> is success for trustpoint <trustpoint_name> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:<hex_dump>`	Connection has been successfully established.
Informational	`Aug 12 10:11:12:I: PKI: enrollment req PKCSREQ/GETCERTINITIAL sent for trust point :< trustpoint_name> to CA :< hostname> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:<hex_dump>`	All enrollment request packets are logged. The message includes the trust point name, CA name, and request type.
Informational	`Aug 12 10:11:12:I: PKI: valid/ invalid pki enrollment response received for trust point:<trustpoint_name> pki status: success/pending : enrollment status: failure/success. <Failure reason> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:<hex_dump>`	All enrollment response packets are logged. The message indicates whether the enrollment response is valid or invalid for the trust point (by name). For valid responses, the response status is included. If the response status is successful, the enrollment status is also indicated. If enrollment fails, the reason for failure is given.
Informational	`Aug 12 10:11:12:I: PKI: authenticate request sent for trust point :< trustpoint_name> to CA :< hostname> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:<hex_dump>`	Authentication requests sent to CA and RA for certificates are logged.
Informational	`Aug 12 10:11:12:I: PKI: authentication reply for trustpoint:<trustpoint_name> pki authentication success/ failure <failure reason> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:<hex_dump>`	Authentication replies from the CA are logged. The reply includes the CA and RA certificates and the trust point authentication status. If authentication fails, the reason for failure is given.
Informational	`Aug 12 10:11:12 :I: PKI: crl/ocsp req sent for trust point :< trustpoint_name> to CA :< hostname> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:<hex_dump>`	PKI requests for CRL or for OCSP packets are logged (the requests are based on revocation check configuration).
Informational	`Aug 12 10:11:12 :I: PKI:crl/ocsp reply for trustpoint:<trustpoint_name>. Peer certificate <serial number> validation success/failed <failure reason> Event:<event_no>,`	PKI responses to the requests for CRL or for OCSP packets are logged. During this process, the peer certificate is validated and the certificate's revocation status is checked based on the CA or OCSP reply. Validation status is

TABLE 14 FIPS syslog messages (continued)

Message level	Message	Explanation
	`pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:<hex_dump>`	also logged. If validation fails, the reason for failure is given.
Informational	`Aug 12 10:11:12 :I: PKI: certificate request for trustpoint/ peer certificate <http url> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:<hex_dump>`	Requests for certificates from external server are logged. The request includes the server URL.
Informational	`Aug 12 10:11:12 :I: PKI: certificate reply for trustpoint/ peer certificate from <http url> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:<hex_dump>`	Replies from external server to request for certificates are logged. The reply includes the certificate.
Informational	`Aug 12 10:11:12 :I: PKI: certificate request for trustpoint/ peer certificate <http url> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:hex_dump>`	Requests for certificates from external server are logged. The request includes the server URL.
Informational	`Aug 12 10:11:12 :I: PKI: certificate reply for trustpoint/ peer certificate from <http url> Event:<event_no>, pkt:<pkt_no>, pkt_len:<pkt_len>, fragment:<frag_no>: frag_len:<frag_len>:hex_dump>`	Replies from external server to request for certificates are logged. The reply includes the certificate.
Informational	`Aug 12 10:11:12 :I: PKI: close connection request for trustpoint to host :< host_name>`	Requests to close the connection between the trust point and the host are logged. The name of the host is included in the log.
Audit events		
Informational	`Dec 16 12:12:29:I:list 102 denied tcp 10.10.10.1(1024)(Ethernet 3/1 0000.0000.0010)`	Deny ACL log file.

Informational	`SYSLOG: <14>Dec 11 2015 17:55:06 mlx32-b Ethernet Port 23/2 has exceeded unknown-unicast high-watermark. Action = alarm-raised, state = UP`
	`SYSLOG: <14>Dec 11 2015 17:55:06 mlx32-b Ethernet port 23/2 is blocked due to unknown-unicast rate-limit threshold auto-shutdown: action = Disabled, state = Down`
	`SYSLOG: <14>Dec 11 2015 17:55:06 mlx32-b Manual intervention is required to bring the port up`
	`SYSLOG: <14>Dec 11 2015 17:55:06 mlx32-b System: Interface ethernet 23/2, state down - shut down by`
		Audit event indicating flooding that has caused packets to be dropped.

TABLE 14 FIPS syslog messages (continued)

Message level	Message	Explanation
	`rate-limiting broadcast, unknown unicast & multicast`	
Informational	`SYSLOG: <14>Dec 16 15:23:06 CER63 Security: console timed out by brocade from USER EXEC mode`	A user on the console was logged out due to inactivity.
Informational	`SYSLOG: <14>Dec 16 15:21:28 CER63 Security: console login by brocade to USER EXEC mode`	A user logged into the console (possibly after session was logged out due to inactivity).

Running Tasks for Different NetIron Devices in FIPS Mode

- MLXe tasks....99
- CER, CER-4X, CER-RT-4X tasks....101

MLXe tasks

The following table lists the running tasks for the Brocade MLXe Series.

TABLE 15 MLXe tasks

Task Name	Priority	Description
idle	0	Idle collector task to find idle CPU usage
con	27	OS console task
mon	31	OS monitor task
flash	20	Flash access task
dbg	30	Debug task
boot	29	Boot task
main	3	Main parent
itc	6	InterTask Communication task
tmr	5	Timer task
ip_rx	5	IP Receive path task
sfm_mgr	9	Switch Fabric Manager task
scp	5	System control task, interacts with modules, ports
lpagent	5	Communicates with various interface modules
console	5	Console task
vlan	5	VLAN handler task
mac_mgr	5	MAC manager task
mrp	5	Metro Ring Protocol handler task
vsrp	5	VSRP handler task
erp	5	ERP handler task
mxrp	5	MSRP handler task
snms	5	Aggregator task for AAA, Syslog, Trap, LLDP, FDP,
rtm	5	Route Table Manager task
rtm6	5	IPv6 Route Table Manager task
ip_tx	5	IP Transmit path task
rip	5	RIP handler task
l2vpn	5	L2VPN (VLL, VLL-Local, VPLS) handler task
mpls	5	MPLS protocol manager
nht	5	Next Hop Table manager task
mpls_glue	5	MPLS task to communicate with internal engine
bgp	5	BGP handler task

TABLE 15 MLXe tasks (continued)

Task Name	Priority	Description
bgp_io	5	BGP I/O controller task
ospf	5	OSPF protocol handler task
ospf_r_calc	5	OSPF route calculation task
isis	5	IS-IS protocol task
isis_spf	5	IS-IS Shortest Path Forward (SPF) calculation task
mcast	5	Multicast protocol task
msdp	5	MSDP manager
vrrp	5	VRRP manager
ripng	5	RIP for IPv6 manager task
ospf6	5	OSPF for IPv6 manager task
ospf6_rt	5	OSPF for IPv6 route calculation task
mcast6	5	Multicast for IPv6 manager task
vrrp6	5	VRRP for IPv6 manager task
bfd	5	Bidirection Fault Detection (BFD) protocol manager task
ipsec	5	IPsec protocol manager task
l4	5	L4 (ACL, Rate Limit) manager task
stp	5	Spanning Tree Protocol (STP) manager task
gvrp_mgr	5	GVRP manager task
snmp	5	SNMP manager task
rmon	5	RMON SNMP table task
web	5	HTTP and HTTPS (SSL/TLS) server and client task
lacp	5	LACP manager task
dot1x	5	802.1X protocol manager task
dot1ag	5	802.1ag protocol manager task
loop_detect	5	L2 loop detection task
ccp	5	MCT Cluster Communication Protocol (CCP): P2P sync between peers
cluster_mgr	5	MCT FSM manager for client and peers
hqos	5	Hierarchical QoS manager task
statistcs	5	Statistics collector manager
hw_access	5	Task doing periodic polling of the temperature sensors
sfm_mon	8	Soft error-related monitoring on SFMs (only applicable certain part number of hSFM)
ntp	5	Network Time Protocol (NTP) manager task
openflow_ofm	5	OpenFlow flow manager task
openflow_opm	5	Openflow protocol manager task
dhcp6	5	DHCP for IPv6 manager task
fid_mgr	5	FID manager
sysmon	5	System monitor task
ospf_msg_task	6	OSPF message handler task
ssl	5	SSL client task
ssh_0	5	SSH client #1, maximum of 16 clients are allowed

CER, CER-4X, CER-RT-4X tasks

The following table lists the running tasks for the Brocade NetIron CER Series.

TABLE 16 CER, CER-4X and CER-RT-4X tasks

Task Name	Priority	Description
idle	0	Idle collector task to find idle CPU usage
con	27	OS console task
mon	31	OS monitor task
flash	20	Flash access task
dbg	30	Debug task
boot	29	Boot task
main	3	Main parent
itc	6	InterTask Communication task
tmr	5	Timer task
ip_rx	5	IP Receive path task
scp	5	System control task, interacts with modules, ports
lpagent	5	Communicates with various interface modules
console	5	Console task
vlan	5	VLAN handler task
mac_mgr	5	MAC manager task
mrp	5	Metro Ring Protocol handler task
vsrp	5	VSRP handler task
erp	5	ERP handler task
mxrp	5	MSRP handler task
snms	5	Aggregator task for AAA, Syslog, Trap, LLDP, FDP, CDP
rtm	5	Route Table Manager task
rtm6	5	IPv6 Route Table Manager task
ip_tx	5	IP Transmit path task
rip	5	RIP handler task
l2vpn	5	L2VPN (VLL, VLL-Local, VPLS) handler task
mpls	5	MPLS protocol manager
nht	5	Next Hop Table manager task
mpls_glue	5	MPLS task to communicate with internal engine
bgp	5	BGP handler task
bgp_io	5	BGP I/O controller task
ospf	5	OSPF protocol handler task
ospf_r_calc	5	OSPF route calculation task
isis	5	IS-IS protocol task
isis_spf	5	IS-IS Shortest Path Forward (SPF) calculation task
mcast	5	Multicast protocol task
msdp	5	MSDP manager
vrrp	5	VRRP manager
ripng	5	RIP for IPv6 manager task

TABLE 16 CER, CER-4X and CER-RT-4X tasks (continued)

Task Name	Priority	Description
ospf6	5	OSPF for IPv6 manager task
ospf6_rt	5	OSPF for IPv6 route calculation task
mcast6	5	Multicast for IPv6 manager task
vrrp6	5	VRRP for IPv6 manager task
bfd	5	Bidirection Fault Detection (BFD) protocol manager task
ipsec	5	IPsec protocol manager task
l4	5	L4 (ACL, Rate Limit) manager task
stp	5	Spanning Tree Protocol (STP) manager task
gvrp_mgr	5	GVRP manager task
snmp	5	SNMP manager task
rmon	5	RMON SNMP table task
web	5	SSL/TLS client task
lacp	5	LACP manager task
dot1x	5	802.1X protocol manager task
dot1ag	5	802.1ag protocol manager task
loop_detect	5	L2 loop detection task
ccp	5	MCT Cluster Communication Protocol (CCP): P2P sync between peers
cluster_mgr	5	MCT FSM manager for client and peers
hw_access	5	HW access manager task
ntp	5	Network Time Protocol (NTP) manager task
openflow_ofm	5	OpenFlow flow manager task
openflow_opm	5	Openflow protocol manager task
dhcp6	5	DHCP for IPv6 manager task
sysmon	5	System monitor task
ospf_msg_task	6	OSPF message handler task
ssl	5	SSL client task
lp	3	Virtual LP (VLP) task
LP-I2C	3	Virtual LP (VLP) I2C bus controller access task
ssh_0	5	SSH client #1, maximum of 16 clients are allowed

OpenSSL License

- OpenSSL license overview.....103

OpenSSL license overview

NOTE

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

NOTE

OpenSSL has been compiled without the Heartbeat extension.

License

This is a copy of the current LICENSE file inside the CVS repository.

```
LICENSE ISSUES
==============
    The OpenSSL toolkit stays under a dual license, i.e. both the conditions of
    the OpenSSL License and the original SSLeay license apply to the toolkit.
    See below for the actual license texts. Actually both licenses are BSD-style
    Open Source licenses. In case of any license issues related to OpenSSL
    please contact openssl-core@openssl.org.

OpenSSL License
---------------

/* ====================================================================
 * Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1.Redistributions of source code must retain the above copyright
 *         notice, this list of conditions and the following disclaimer.
 *
 * 2.Redistributions in binary form must reproduce the above copyright
 *         notice, this list of conditions and the following disclaimer in
 *         the documentation and/or other materials provided with the
 *         distribution.
 *
 * 3.All advertising materials mentioning features or use of this
 *         software must display the following acknowledgment:
 *         "This product includes software developed by the OpenSSL Project
 *         for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4.The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *         endorse or promote products derived from this software without
 *         prior written permission. For written permission, please contact
 *         openssl-core@openssl.org.
 *
 * 5.Products derived from this software may not be called "OpenSSL"
 *         nor may "OpenSSL" appear in their names without prior written
 *         permission of the OpenSSL Project.
 *
 * 6.Redistributions of any form whatsoever must retain the following acknowledgment:
 *         "This product includes software developed by the OpenSSL Project
 *         for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
```

```
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
* ====================================================================
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License
-----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1.Redistributions of source code must retain the copyright
 *         notice, this list of conditions and the following disclaimer.
 * 2.Redistributions in binary form must reproduce the above copyright
 *         notice, this list of conditions and the following disclaimer in the
 *         documentation and/or other materials provided with the distribution.
 * 3.All advertising materials mentioning features or use of this software
 *         must display the following acknowledgement:
 *         "This product includes cryptographic software written by
 *         Eric Young (eay@cryptsoft.com)"
 *         The word 'cryptographic' can be left out if the rouines from the library
 *         being used are not cryptographic related :-).
 * 4.If you include any Windows specific code (or a derivative thereof) from
 *         the apps directory (application code) you must include an acknowledgement:
 *         "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
```

```
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/
```