Configuring FIPS Mode

The Federal Information Processing Standard (FIPS) 140-2 is an U.S. and Canadian government certification standard that defines requirements that the cryptographic modules must follow. The FIPS specifies best practices for implementing cryptographic algorithms, handling key material and data buffers, and working with the operating system.

In Cisco IOS XR software, these applications are verified for FIPS compliance:

- Secure Shell (SSH)
- Secure Socket Layer (SSL)
- Transport Layer Security (TLS)
- Internet Protocol Security (IPSec) for Open Shortest Path First version 3 (OSPFv3)
- Simple Network Management Protocol version 3 (SNMPv3)
- AAA Password Security

Note Any process that uses any of the following cryptographic algorithms is considered non-FIPS compliant:

- Rivest Cipher 4 (RC4)
- Message Digest (MD5)
- Keyed-Hash Message Authentication Code (HMAC) MD5
- Data Encryption Standard (DES)

The Cisco Common Cryptographic Module (C3M) provides cryptographic services to a wide range of the networking and collaboration products of Cisco. This module provides FIPS-validated cryptographic algorithms for services such as RTP, SSH, TLS, 802.1x, and so on. The C3M provides cryptographic primitives and functions for the users to develop any protocol.

By integrating with C3M, the Cisco IOS-XR software is compliant with the FIPS 140-2 standards and can operate in FIPS mode, level 1 compliance.

Prerequisites for Configuring FIPS

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command.

If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

How to Configure FIPS

Perform these tasks to configure FIPS.

Enable FIPS mode

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **crypto fips-mode**

Example:

```
Router(config)#crypto fips-mode
```

Enters FIPS configuration mode.

Note Stop new incoming SSH sessions while configuring or unconfiguring **crypto fips-mode**. Restart the router upon configuration.

Step 3 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.
- **No** —Exits the configuration session without committing the configuration changes.
- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Step 4 **show logging**

Example:

```
Router#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level debugging, 60 messages logged
```

```
    Monitor logging: level debugging, 0 messages logged
    Trap logging: level informational, 0 messages logged
    Buffer logging: level debugging, 3 messages logged

Log Buffer (9000000 bytes):
<output omitted>

Log Buffer (307200 bytes):

RP/0/RSP0/CPU0:Apr 16 12:48:17.736 : cepki[433]: The configuration setting for FIPS mode has been
modified. The system must be reloaded to finalize this configuration change. Please refer to the IOS
 XR System Security Configuration Guide, Federal Information Process Standard(FIPS) Overview section
 when modifying the FIPS mode setting.
RP/0/RSP0/CPU0:Apr 16 12:48:17.951 : config[65757]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'lab'. Use 'show configuration commit changes 1000000002' to view
the changes.
RP/0/RSP0/CPU0:Apr 16 12:48:23.988 : config[65757]: %MGBL-SYS-5-CONFIG_I : Configured from console
by lab


....
....
....
```

Displays the contents of logging buffers.

Note Use the **show logging | i fips** command to filter FIPS specific logging messages.

Step 5 **reload location all**

Example:

```
Router#reload location all
```

Reloads a node or all nodes on a single chassis or multishelf system.

Configure FIPS-compliant Keys

Perform these steps to configure the FIPS-compliant keys:

Note The crypto keys are auto-generated at the time of router boot up. You need to perform these steps to generate the keys only if the keys are missing under some scenarios.

Before you begin

Refer the configuration steps in the Enable FIPS mode, on page 2 section for enabling the FIPS mode.

Step 1 **crypto key generate rsa [usage-keys | general-keys]** *key label*

Example:

```
Router#crypto key generate rsa general-keys rsakeypair
```

Generate a RSA key pair. Ensure that all the key pairs meet the FIPS requirements. The RSA key sizes allowed under FIPS mode are 2048, 3072 and 4096.

The option **usage-keys** generates separate RSA key pairs for signing and encryption. The option **general-keys** generates a general-purpose RSA key pair for signing and encryption.

To delete the RSA key pair, use the **crypto key zeroize rsa** *keypair-label* command.

Step 2 **crypto key generate dsa**

Example:

```
Router#crypto key generate dsa
```

Generate a DSA key pair if required. Ensure that all the key pairs meet the FIPS requirements. The DSA key size allowed under FIPS mode is 2048.

To delete the DSA key pair, use the **crypto key zeroize dsa** *keypair-label* command.

Step 3 **crypto key generate ecdsa**

Example:

```
Router#crypto key generate ecdsa
```

Generate a ECDSA key pair if required. Ensure that all the key pairs meet the FIPS requirements. The ECDSA key sizes allowed under FIPS mode are **nistp256**, **nistp384** and **nistp512**.

To delete the DSA key pair, use the **crypto key zeroize ecdsa** *keypair-label* command.

Step 4 **show crypto key mypubkey rsa**

Example:

```
Router#show crypto key mypubkey rsa
```

Displays the existing RSA key pairs

Step 5 **show crypto key mypubkey dsa**

Example:

```
Router#show crypto key mypubkey dsa
```

Displays the existing DSA key pairs

Configure FIPS-compliant Key Chain

Perform these steps to configure the FIPS-compliant key chain:

Before you begin

Refer the configuration steps in the Enable FIPS mode, on page 2 section for enabling the FIPS mode.

Step 1 **configure**

Example:

```
Router#configure
```

Enters the global configuration mode.

Step 2 **key chain** *key-chain-name*

Example:

```
Router(config)#key chain mykeychain
```

Creates a key chain.

Step 3 **key** *key-id*

Example:

```
Router(config-mykeychain)#key 1
```

Creates a key in the key chain.

Step 4 **cryptographic-algorithm** {**HMAC-SHA1-20** | **SHA-1**}

Example:

```
Router(config-mykeychain-1)#cryptographic-algorithm HMAC-SHA1-20
```

Configures the cryptographic algorithm for the key chain. Ensure that the key chain configuration always uses SHA-1 as the hash or keyed hash message authentication code (hmac) algorithm.

Step 5 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.
- **No** —Exits the configuration session without committing the configuration changes.
- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configure FIPS-compliant Certificates

Perform these steps to configure the FIPS-compliant certificates:

Before you begin

Refer the configuration steps in the Enable FIPS mode, on page 2 section for enabling the FIPS mode.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **crypto ca trustpoint** *ca-name key label*

Example:

```
Router(config)#crypto ca trustpoint msiox rsakeypair
```

Configures the trustpoint by utilizing the desired RSA keys.

Ensure that the certificates meet the FIPS requirements for key length and signature hash or encryption type.

Note The minimum key length for RSA or DSA key is 1024 bits. The required hash algorithm is SHA-1-20.

Step 3 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.
- **No** —Exits the configuration session without committing the configuration changes.
- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Step 4 **show crypto ca certificates**

Example:

```
Router#show crypto ca certificates
```

Displays the information about the certificate

What to do next

For more information about certification authority and requesting router certificates, see the *Implementing Certification Authority* chapter in this guide.

Configure FIPS-compliant OSPFv3

Perform these steps to configure the FIPS-compliant OSPFv3:

Before you begin

Refer the configuration steps in the Enable FIPS mode, on page 2 section for enabling the FIPS mode.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **router ospfv3** *process name*

Example:

```
Router(config)#router ospfv3 ospfname
```

Configures the OSPFv3 process.

Step 3 **area** *id*

Example:

```
Router(config-ospfv3)#area 1
```

Configures the OSPFv3 area ID. The ID can either be a decimal value or an IP address.

Step 4 **authentication** {**disable** | **ipsec spi** *spi-value* **sha1** [**clear** | **password**] *password*}

Example:

```
Router(config-ospfv3-ar)#authentication ipsec spi 256 sha1 password pa1
```

Enables authentication for OSPFv3. Note that the OSPFv3 configuration supports only SHA-1 for authentication.

Note IPSec is supported only for Open Shortest Path First version 3 (OSPFv3).

Step 5 **exit**

Example:

```
Router(config-ospfv3-ar)#exit
```

Exits OSPFv3 area configuration and enters the OSPFv3 configuration mode.

Step 6 **encryption** {**disable** | {**ipsec spi** *spi-value* **esp** {**3des** | **aes** [**192** | **256**] [**clear** |**password**] *encrypt-password*} [**authentication sha1** [**clear** | **password**] *auth-password*]}}

Example:

```
Router(config-ospfv3)#encryption ipsec spi 256 esp 3des password pwd
```

Encrypts and authenticates the OSPFv3 packets. Ensure that the OSPFv3 configuration uses the following for encryption in the configuration.

- 3DES: Specifies the triple DES algorithm.
- AES: Specifies the Advanced Encryption Standard (AES) algorithm.

Ensure that SHA1 is chosen if the authentication option is specified.

Step 7 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.
- **No** —Exits the configuration session without committing the configuration changes.
- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configure FIPS-compliant SNMPv3 Server

Perform these steps to configure the FIPS-compliant SNMPv3 server:

Before you begin

Refer the configuration steps in the Enable FIPS mode, on page 2 section for enabling the FIPS mode.

Step 1 **configure**

Example:

```
Router#configure
```

Enters the global configuration mode.

Step 2 **snmp-server user** *username groupname* {**v3** [**auth sha** {**clear** | **encrypted**} *auth-password* [**priv** {**3des** |**aes** { **128** | **192** | **256**} } {**clear** | **encrypted** } *priv-password*]] } [**SDROwner** | **SystemOwner**] *access-list-name*

Example:

```
Router(config)#snmp-server user user1 g v3 auth sha clear pass priv aes 128 clear privp
```

Configures the SNMPv3 server.

Step 3 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.
- **No** —Exits the configuration session without committing the configuration changes.
- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configure FIPS-compliant SSH Client and Server

Perform these steps to configure the FIPS-compliant SSH Client and the Server:

Before you begin

Refer the configuration steps in the Enable FIPS mode, on page 2 section for enabling the FIPS mode.

Step 1 **ssh** {*ipv4-address* | *ipv6-address*} **cipher aes** {**128-CTR** | **192-CTR** | **256-CTR**} **username** *username*

Example:

```
Router#ssh 192.0.2.1 cipher aes 128-CTR username user1
```

Starts an SSH session to the server using the FIPS-approved ciphers. Ensure that the SSH client is configured only with the FIPS-approved ciphers. AES(Advanced Encryption Standard)-CTR (Counter mode) is the FIPS-compliant cipher algorithm with key lengths of 128, 192 and 256 bits.

Step 2 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 3 **ssh server v2**

Example:

```
Router(config)#ssh server v2
```

Configures the SSH server.

The supported key exchange algorithms are:

- diffie-hellman-group14-sha1
- ecdh-sha2-nistp256
- ecdh-sha2-nistp384
- ecdh-sha2-nistp521

The supported cipher algorithms are:

- aes128-ctr
- aes192-ctr
- aes256-ctr
- aes128-gcm
- aes256-gcm

The supported HMAC algorithms are:

- hmac-sha2-512
- hmac-sha2-256
- hmac-sha1

Step 4 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.
- **No** —Exits the configuration session without committing the configuration changes.
- **Cancel** —Remains in the configuration session, without committing the configuration changes.