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, on page 2
- How to Configure FIPS, on page 2

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.

Guidelines for Enabling FIPS Mode

From Cisco IOS XR Software Release 7.2.1 and later, you must follow these guidelines while enabling FIPS mode:

- You must configure the session with a FIPS-approved cryptographic algorithm. A session configured with non-approved cryptographic algorithm for FIPS (such as, **MD5** and **HMAC-MD5**) does not work. This is applicable for OSPF, BGP, RSVP, ISIS, or any application using key chain with non-approved cryptographic algorithm, and only for FIPS mode (that is, when **crypto fips-mode** is configured).
- If you are using any **HMAC-SHA** algorithm for a session, then you must ensure that the configured *key-string* has a minimum length of 14 characters. Otherwise, the session goes down. This is applicable only for FIPS mode.
- If you try to execute the telnet configuration on a system where the FIPS mode is already enabled, then the system rejects the telnet configuration.
- If telnet configuration already exists on the system, and if FIPS mode is enabled later, then the system rejects the telnet connection. But, it does not affect the telnet configuration as such.
- It is recommended to configure the **crypto fips-mode** command first, followed by the commands related to FIPS in a separate commit. The list of commands related to FIPS with non-approved cryptographic algorithms are:
  - **key chain** *key-chain-name* **key** *key-id* **cryptographic-algorithm MD5**
  - **key chain** *key-chain-name* **key** *key-id* **cryptographic-algorithm HMAC-MD5**
  - **router ospfv3 1 authentication ipsec spi 256 md5** *test-md5-value*
  - **router ospfv3 1 encryption ipsec spi 256 esp des** *test-des-value*
  - **router ospfv3 1 encryption ipsec spi 256 esp des** *test-des-value* **authentication md5** *test-md5-value*
  - **snmp-server user** *user1 user-grp1* **v3 auth md5 priv des56**
  - **ssh server algorithms key-exchange diffie-hellman-group1-sha1**
  - **telnet vrf default ipv4 server max-servers** *100*

How to Configure FIPS

Perform these tasks to configure FIPS.

Enable FIPS mode

Procedure

Step 1 **configure**

Example:

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

Enters 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.

Note From Cisco IOS XR Release 7.3.2 onwards, you can generate and delete key-pairs from XR Config mode. For more details, see Public Key-Pair Generation in XR Config Mode in the chapter *Implementing Certification Authority Interoperability* .

Before you begin

Refer the configuration steps in the Enable FIPS mode, on page 3 section for enabling the FIPS mode.

Procedure

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
Fri Mar 27 14:00:20.954 IST
Key label: system-root-key
Type : RSA General purpose
Size : 2048
Created : 01:13:10 IST Thu Feb 06 2020
Data :
 30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
 00A93DE0 1E485EE3 0E7F0964 C48361D1 B6014BE7 A303D8D6 F7790E92 88E69C4B
 B97B7A9C D1B277E3 1569093C 82BD3258 7F67FB49 94860ECD 34498F1F 59B45757
 F32C8E8F 7CEE23EC C36A43D1 9F85C0D9 B96A14DD DD3BBD4C A1FB0888 EED210A7
 39D9A403 7ACE0F6E 39107226 CA621AD8 6E8102CA 9761B86F D33F2871 9DD16559
 AFCB4729 EFCEDBAF 83DF76E4 9A439844 EE3B1180 4022F575 99E11A2C E25BB23D
 9DD74C81 4E5C1345 D9E3CC79 1B98B1AA 6C06F004 22B901EC 36C099FE 10DE2622
 EB7CE618 9A555769 12D94C90 D9BEE5EA A664E7F6 4DF8D8D4 FE7EAB07 1EF4FEAB
 22D9E55F 62BA66A0 72153CEC 81F2639F B5F2B5C5 25E10364 19387C6B E8DB8990
 11020301 0001
Key label: system-enroll-key
Type : RSA General purpose
Size : 2048
Created : 01:13:16 IST Thu Feb 06 2020
Data :
 30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
 009DBC14 C83604E4 EB3D3CF8 5BA7FDDB 80F7E85B 427332D8 BBF80148 F0A9C281
 49F87D5C 0CEBA532 EBE797C5 7F174C69 0735D13A 493670CB 63B04A12 4BCA7134
 EE0031E9 047CAA1E 802030C5 6071E8C2 F8ECE002 CC3B54E7 5FD24E5C 61B7B7B0
 68FA2EFA 0B83799F 77AE4621 435D9DFF 1D713108 37B614D3 255020F9 09CD32E8
 82B07CD7 01A53896 6DD92B5D 5119597C 98D394E9 DBD1ABAF 6DE949FE 4A8BF1E7
 851EB3F4 60B1114A 1456723E 063E50C4 2D410906 BDB7590B F1D58480 F3FA911A
 6C9CD02A 58E68D04 E94C098F 0F0E81DB 76B40C55 64603499 2AC0547A D652412A
 BCBBF69F 76B351EE 9B2DF79D E490C0F6 92D1BB97 B905F33B FAB53C20 DDE2BB22
 C7020301 0001
```

Displays the existing RSA key pairs.

You can also view the RSA keys in the running configuration. The keys in the following example are in OpenSSL format.

Note
Only those keys that are generated in the `config` mode are visible in the running configuration.

```
Router(config)#crypto key generate rsa test
Router(config)#commit
Thu May 12 08:37:59.894 UTC
Router(config)#end
Router#show running-config
Thu May 12 08:38:04.244 UTC
Building configuration...
!! IOS XR Configuration 7.3.4
!! Last configuration change at Thu May 12 08:37:59 2022 by cisco
!
username cisco
 group root-lr
 group cisco-support
 secret 10
$6$8zR0nTbkA7A1n...$0Kn.YxNNmh1cXo9cEvEwLGAFf.rEOTycjsizI/TLBz9WoQX.rmxVwkNgTKAnROUGPtBVlQ/Ndew8gEREXJ7mI0
!
call-home
 service active
 contact smart-licensing
 profile CiscoTAC-1
  active
  destination transport-method http
 !
!
interface MgmtEth0/RSP0/CPU0/0
 shutdown
!
crypto key generate rsa test general-keys 2048 | -----BEGIN PUBLIC KEY-----
MIIBIDANBgkqhkiG9w0BAQEFAAOCAQ0AMIIBCAKCAQEAgiXFnld/AADcil6eV38A
AIIlxZ5XfwAAcJb6e1d/AAAA7du+AAAAAI6Qs47BQLhIVQAAAAAAAAAAWQDQVn8A
ANyKXp5XfwAAKAAAAAAAAACaNcWeV38AANyKXp5XfwAAmjXFnld/AADcil6eV38A
AJo1xZ5XfwAAAO3bvgAAAABVAAAAAAAAABBEANBWfwAA3Ipenld/AAAgAAAAAAAA
AI8lxZ5XfwAA3Ipenld/AACPJcWeV38AAHhZANBWfwAAAO3bvgAAAADUTNDpQMWp
UUUAAAAAAAAAkBcA0FZ/AADcil6eV38AABgAAAAAAAAiSXFnld/AADcil6eV38A
AAIBAA==
-----END PUBLIC KEY-----
 |
end
```

Step 5 **show crypto key mypubkey dsa**

Example:

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

Displays the existing DSA key pairs.

You can also view the DSA key in the running configuration. The keys in the following example are in OpenSSL format:

Note
Only those keys that are generated in the `config` mode are visible in the running configuration.

```
Router(config)#crypto key generate dsa
Router(config)#commit
Thu May 12 08:39:32.093 UTC
Router(config)#end
Router#show running-config
Thu May 12 08:39:37.557 UTC
```

```
Building configuration...
!! IOS XR Configuration 7.3.4
!! Last configuration change at Thu May 12 08:39:32 2022 by cisco
!
username cisco
 group root-lr
 group cisco-support
 secret 10
$6$8zR0nTbkA7A1n...$0Kn.YxNNmh1cXo9cEvEwLGAFf.rEOTycjsizI/TLBz9WoQX.rmxVwkNgTKAnROUGPtBVlQ/Ndew8gEREXJ7mI0
!
call-home
 service active
 contact smart-licensing
 profile CiscoTAC-1
  active
  destination transport-method http
 !
!
interface MgmtEth0/RSP0/CPU0/0
 shutdown
!
crypto key generate dsa 1024 | -----BEGIN PUBLIC KEY-----
MIIBcTCB8AYHKoZIzjgEATCB5AKBgQCPJcWeV38AANyKXp5XfwAAjyXFnld/AADI
CQDQVn8AAADt274AAAAA2hu9QE4nZsllAAAAAAAAADAXANBWfwAA3Ipenld/AAAg
AAAAAAAAAIIlxZ5XfwAA3Ipenld/AACCJcWeV38AACtah6B+cCkKAO3bvgAAAADc
gpej73WrUwIUSD4A0FZ/AAAQAAAAEAAAAAAAAAACSAEAAAAAAAAAAAAAAAAAAAAB
AAAAAAAAAAAAAAAAAAAAAYJtQfFd/AAAAAAAAAAAAACgzANBWfwAAAO3bvgAAAAC2
3xsW4Kjy1wN8AAJ5EAAAABAAAH9W0AA+SAAAAAAA7du+AAAAAFUAAAAAAAAoDEA
0FZ/AAAV8DieV38AACAAAAAAAAAAHvE4nld/AAAV8DieV38AAB7xOJ5XfwAAAAAA
AAAAAAAA7du+AAAAAJEOuuQWSrUHVQAAAAAAAAAAAAAAAAAAAAAA==
-----END PUBLIC KEY-----
 |
crypto key generate rsa test general-keys 2048 | -----BEGIN PUBLIC KEY-----
MIIBIDANBgkqhkiG9w0BAQEFAAOCAQ0AMIIBCAKCAQEAgiXFnld/AADcil6eV38A
AIIlxZ5XfwAAcJb6e1d/AAAA7du+AAAAAI6Qs47BQLhIVQAAAAAAAAAAWQDQVn8A
ANyKXp5XfwAAKAAAAAAAAACaNcWeV38AANyKXp5XfwAAmjXFnld/AADcil6eV38A
AJo1xZ5XfwAAAO3bvgAAAABVAAAAAAAAABBEANBWfwAA3Ipenld/AAAgAAAAAAAA
AI8lxZ5XfwAA3Ipenld/AACPJcWeV38AAHhZANBWfwAAAO3bvgAAAADUTNDpQMWp
UUUAAAAAAAAAkBcA0FZ/AADcil6eV38AABgAAAAAAAAiSXFnld/AADcil6eV38A
AAIBAA==
-----END PUBLIC KEY-----
 |
end|
```

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 3 section for enabling the FIPS mode.

Procedure

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 3 section for enabling the FIPS mode.

Procedure

Step 1 **configure**

Example:

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

Enters 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 3 section for enabling the FIPS mode.

Procedure

Step 1 **configure**

Example:

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

Enters 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 3 section for enabling the FIPS mode.

Procedure

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 3 section for enabling the FIPS mode.

Procedure

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 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.