Juniper Networks EX4650, QFX5120 and QFX5210 Ethernet Switches with JUNOS 20.2R1-S1

Non-Proprietary FIPS 140-2 Cryptographic Module Security Policy

Version: 1.3
November 8, 2021

Juniper Networks, Inc.

1133 Innovation Way

Sunnyvale, California 94089

USA

408.745.2000

1.888 JUNIPER

www.juniper.net

Table of Contents

List of Tables

List of Figures

1 Introduction

The Juniper Networks QFX series switches are high performance, high density data center switches. The QFX switches provide high performance, wire speed switching with low latency and jitter. The QFX series switches provide the universal building blocks for multiple data center fabric architectures.

This Security Policy covers the following Ethernet switch models:

- QFX5120-48T
- QFX5120-48Y
- QFX5120-32C
- QFX5210-64C
- EX4650-48Y

This is a non-proprietary Cryptographic Module Security Policy for the Juniper Networks EX4650, QFX5120 and QFX5210 Ethernet switches cryptographic module from Juniper Networks, hereafter referred to as the module. It provides detailed information relating to each of the FIPS 140-2 security requirements relevant to Juniper Networks EX4650, QFX5120 and QFX5210 Ethernet switches module along with instructions on how to run the module in a secure FIPS 140-2 mode.

All four models run Juniper's Junos OS firmware. The validated version of the firmware is Junos OS 20.2R1-S1. The names of the image files are:

- jinstall-host-qfx-5e-x86-64-20.2R1-S1.5-secure-signed.tgz (for all QFX platforms)
- jinstall-host-ex-4e-x86-64-20.2R1-S1.5-secure-signed.tgz (only for EX platforms)

The module is defined as a multiple-chip standalone module that execute Junos OS 20.2R1-S1 firmware on the switch models listed in Table 1. The cryptographic boundary is defined as the outer edge of the switch. The module's operational environment is a non-modifiable operational environment.

Table 1 provides a list of the hardware versions that are part of the module validation and the basic configuration of the hardware.

Table 1 – Cryptographic Module Configurations

Model	Hardware Versions	Chassis differences	Network Ports
QFX5120-48T	QFX5120-48T-AFI QFX5120-48T-AFO QFX5120-48T-DC-AFI QFX5120-48T-DC-AFO	AC airflow in AC airflow out DC airflow in DC airflow out	48x10GbE+6x100GbE or 48x25GbE+6x100GbE
QFX5120-48Y	QFX5120-48Y-AFI2 QFX5120-48Y-AFO2 QFX5120-48Y-DC-AFI2 QFX5120-48Y-DC-AFO2	AC Unit Air flow in AC Unit Air flow out DC Unit Air flow in DC Unit Air flow out	48x10GbE + 8x40GbE or 48x25GbE + 8x100GbE

QFX 5120-32C	QFX5120-32C-AFI QFX5120-32C-AFO QFX5120-32C-DC-AFI QFX5120-32C-DC-AFO	AC Unit Air flow in AC Unit Air flow out DC Unit Air flow in DC Unit Air flow out	32x100GbE
QFX5210-64C	QFX5210-64C-AFI QFX5210-64C-AFO QFX5210-64C-DC-AFI QFX5210-64C-DC-AFO	AC Unit Air flow in AC Unit Air flow out DC Unit Air flow in DC Unit Air flow out	64 QSFP+/QSFP28 ports
EX4650-48Y	EX4650-48Y-AFI EX4650-48Y-AFO EX4650-48Y-DC-AFI EX4650-48Y-DC-AFO	AC Unit Air flow in AC Unit Air flow out DC Unit Air flow in DC Unit Air flow out	48x25GbE/10GbE/GbE SFP28/SFP+/SFP ports, 8x100GbE/40GbE QSFP28/QSFP+ ports

The module is designed to meet FIPS 140-2 Level 1 overall:

Table 2 – Security Level of Security Requirements

Area	Description	Level
1	Module Specification	1
2	Ports and Interfaces	1
3	Roles and Services	3
4	Finite State Model	1
5	Physical Security	1
6	Operational Environment	N/A
7	Key Management	1
8	EMI/EMC	1
9	Self-test	1
10	Design Assurance	3
11	Mitigation of Other Attacks	N/A
Overall		1

The module has a non-modifiable operational environment as per the FIPS 140-2 definitions. It includes a firmware load service to support necessary updates. New firmware versions within the scope of this validation must be validated through the FIPS 140-2 CMVP. Any other firmware loaded into the module is out of the scope of this validation and require a separate FIPS 140-2 validation.

The module does not implement any mitigations of other attacks as defined by FIPS 140-2.

Juniper's development processes are such that future releases of Junos should be FIPS validate-able when run on the same hardware platform and meet the claims made in this document. Only the versions that explicitly appear on the certificate, however, are formally validated. The CMVP makes no claim as to the correct operation of the module or the security strengths of the generated keys when operating under a version that is not listed on the validation certificate.

1.1 Hardware and Physical Cryptographic Boundary

The physical forms of the module are depicted in Figure 3 to Figure 10. The module is completely enclosed in a rectangular nickel or clear zinc coated, cold rolled steel, plated steel and brushed aluminum enclosure. For all models, the cryptographic boundary is defined as the outer edge of the switch chassis. The module does not rely on external devices for input and output of critical security parameters (CSPs).

Figure 1 - QFX 5120-48T front view

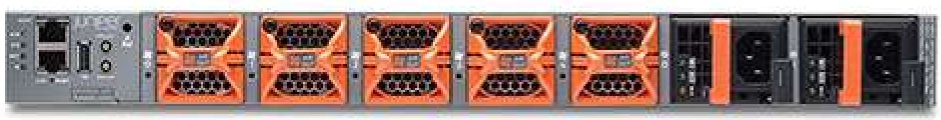

Figure 2 - QFX 5120-48T rear view

Figure 3 - QFX 5120-48Y front view

Figure 4 - QFX 5120-48Y rear view

Figure 5 - QFX 5120-32C front view

Figure 6 - QFX 5120-32C rear view

Figure 7 - QFX 5210-64C front view

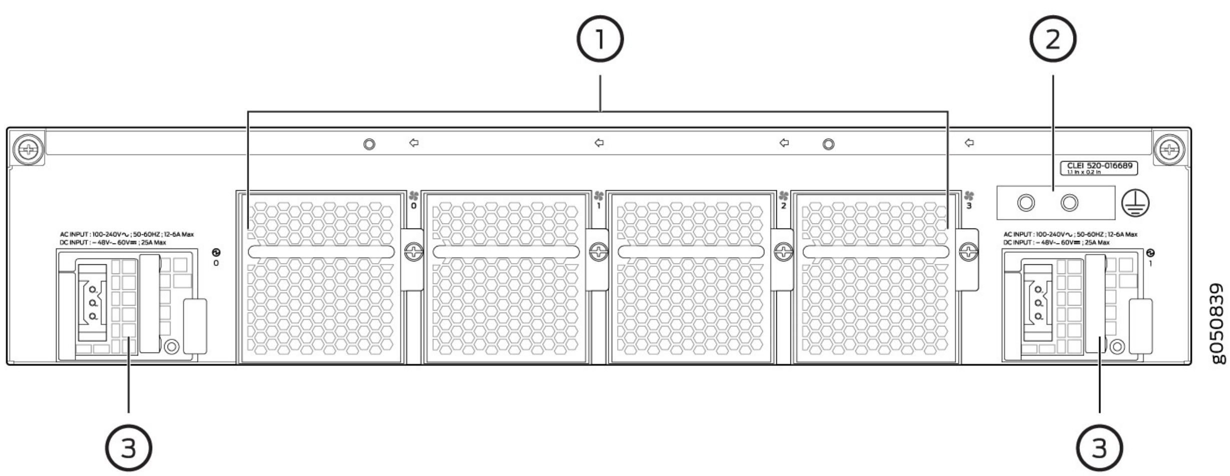

Figure 8 - QFX 5210-64C rear view

Figure 9 - EX 4650-48Y front view

Figure 10 - EX4650-48Y rear view

The following table maps each logical interface type defined in the FIPS 140-2 standard to one or more physical interfaces.

Table 3 – Port and Interface types

Port	Description	Logical Interface Type
Ethernet	LAN Communications	Control in, Data in, Data out, Status out
Serial	Console serial port	Control in, Data in, Data out, Status out
MGMT	Out-of-band management port	Control in, Data in, Data out, Status out
Power	Power connector	Power in
Reset	Reset button	Control in
LED	Status indicator lighting	Status out
USB	Firmware load port	Control in, Data in

The following table provides a detailed description of the ports and interfaces available for each model.

Table 4 – Ports and Interfaces

Router model	Power supply port	Fan modules	Console port	Management port	USB port	Built-In Ports	Pluggable
QFX5120-48T-AFI	2	5	1	1	1	48	6 SFP ports
QFX5120-48T-AFO	2	5	1	1	1	48	6 SFP ports
QFX5120-48T-DC-AFI	2	5	1	1	1	48	6 SFP ports
QFX5120-48T-DC-AFO	2	5	1	1	1	48	6 SFP ports
QFX5120-48Y-AFI2	2	5	1	2	1	0	56 SFP ports
QFX5120-48Y-AFO2	2	5	1	2	1	0	56 SFP ports
QFX5120-48Y-DC-AFI2	2	5	1	2	1	0	56 SFP ports
QFX5120-48Y-DC-AFO2	2	5	1	2	1	0	56 SFP ports
QFX5120-32C-AFI	2	6	1	2	1	0	32 SFP ports
QFX5120-32C-AFO	2	6	1	2	1	0	32 SFP ports
QFX5120-32C-DC-AFI	2	6	1	2	1	0	32 SFP ports
QFX5120-32C-DC-AFO	2	6	1	2	1	0	32 SFP ports
QFX5210- 64C-AFI	2	4	1	2	1	0	64 SFP ports
QFX5210- 64C-AFO	2	4	1	2	1	0	64 SFP ports
QFX5210-64C-DC-AFI	2	4	1	2	1	0	64 SFP ports
QFX5210-64C-DC-AFO	2	4	1	2	1	0	64 SFP ports
EX4650-48Y-AFI	2	5	1	2	1	48	8 SFP ports
EX4650-48Y-AFO	2	5	1	2	1	48	8 SFP ports
EX4650-48Y-DC-AFI	2	5	1	2	1	48	8 SFP ports
EX4650-48Y-DC-AFO	2	5	1	2	1	48	8 SFP ports

1.2 Mode of Operation

The module provides a non-Approved mode of operation in which non-Approved cryptographic algorithms are supported. The module supports non-Approved algorithms when operating in the non-Approved mode of operation as described in Sections 2.4 and 3.4. When transitioning between the non-Approved mode of operation and the Approved mode of operation, the CO must zeroize all CSPs by following the instructions in Section 1.3.

Then, the Cryptographic Officer (CO) must run the following commands to configure the module into the Approved mode of operation:

co@fips-qfx# set system fips level 1

co@fips-qfx# commit

Once the Junos firmware image is installed, configured into Approved mode and rebooted, and integrity and self-tests have run successfully on initial power-on, the module is operating in the Approved mode. This prevents access to non FIPS approved functionality. Transitioning back to non-approved mode is only possible via zeroising the module as described in Section 1.3.

The operator can verify the module is operating in the Approved mode by verifying the following:

- The "show version local" command indicates that the module is running the Approved firmware (i.e. Junos Software Release 20.2R1-S1).
- The command prompt ends in ":fips", which indicates the module has been configured in the Approved mode of operation.

1.3 Zeroization

The following command allows the Cryptographic Officer to zeroize CSPs contained within the module:

co@fips-qfx> request system zeroize

Zeroization completely erases all configuration information on the device, including all cryptographic keys and CSPs and returns the module to its factory default state.

Note: The Cryptographic Officer must retain control of the module while zeroization is in process.

2 Cryptographic Functionality

The module implements the FIPS Approved, vendor affirmed, and non-Approved-but-Allowed cryptographic functions listed in Table 5 through Table 8 below. Table 9 summarizes the high-level protocol algorithm support. Although the module may have been tested for additional algorithms or modes, only those listed below are actually utilized by the module.

2.1 Approved Algorithms

References to standards are given in square bracket [ ]; see the References table.

Table 5 – Kernel Cryptographic Functions

CAVP Cert.	Algorithm	Mode	Key Lengths, Curves, or Moduli	Functions
A866	HMAC [198]	SHA-1	λ = 160	Message Authentication
		SHA-256	λ = 256	
A866	SHS [180]	SHA-1 SHA-256 SHA-384 SHA-512		Message Digest Generation
A866	DRBG [90A]	HMAC	SHA-256	Random Bit Generation

Table 6 – OpenSSL Approved Cryptographic Functions

CAVP Cert.	Algorithm	Mode	Key Lengths, Curves, or Moduli	Functions
A867	AES [197]	CBC [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
		CTR [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
N/A[1]	CKG	[133] Section 6.1		Asymmetric key generation using unmodified DRBG output
A867	CVL	ECC CDH [56A][2]	P-256 (SHA 256) P-384 (SHA 384) P-521 (SHA 512)	ECC CDH primitive used as part of key agreement for SSH protocol
N/A[3]	KAS-SSC [56ARev3]	FFC DH	MODP-2048 (ID=14)	FFC DH primitive used as part of key agreement for SSH protocol
A867	ECDSA [186-4]		P-256 (SHA-256) P-384 (SHA-384) P-521 (SHA-512)	KeyGen, PKV, SigGen, SigVer

[1] Vendor affirmed.

[2] ECDH is only allowed until June 30, 2022.

[3] Vendor affirmed as per IG D.1-rev3

A867	HMAC [198]	SHA-1	λ = 160	SSH Message Authentication DRBG Primitive
		SHA-256	λ = 256	
		SHA-384	λ = 384	
		SHA-512	λ = 512	
A867	RSA [186-4]		n=2048 (SHA 256, 384, 512) n=3072 (SHA 256, 384, 512) n=4096(SHA 256, 384, 512)	SigGen
			n=2048 (SHA 256, 384, 512) n=3072 (SHA 256, 384, 512) n=4096(SHA 256, 384, 512)	SigVer
			n=2048 n=3072 n=4096	KeyGen
A867	SHS [180]	SHA-1 SHA-256 SHA-384 SHA-512		Message Digest Generation, SSH KDF Primitive
A867	Triple-DES[4] [67]	TCBC [38A]	Key Size: 192	Encrypt, Decrypt
A867	DRBG [90A]	HMAC	SHA 256	Random Bit Generation
A867	CVL	SSH [135]	SHA 1, 256, 384, 512	Key Derivation

Table 7 – LibMD Approved Cryptographic Functions

CAVP Cert.	Algorithm	Mode	Key Lengths, Curves, or Moduli	Functions
A868	HMAC [198]	SHA-1	λ = 160	Password Hashing
		SHA-256	λ = 256	
	SHS [180]	SHA-1 SHA-256 SHA-512		Message Digest Generation

4 The module enforces a limit of 2^{20} transforms per Triple-DES key. Use of Triple-DES in this module is only allowed until December 31, 2023.

2.2 Allowed Algorithms

Table 8 – Allowed Cryptographic Functions

Algorithm	Caveat	Use
NDRNG [IG] 7.14 Scenario 1a	The module generates a minimum of 256 bits of entropy for key generation.	Seeding the DRBG
Elliptic Curve Diffie-Hellman [IG] D.8[5]	CVL Cert. #A867 with CVL Cert. #A867; provides between 128 and 256 bits of encryption strength.	SSH key agreement including key derivation

2.3 Protocols

Table 9 – Protocols using approved algorithms in FIPS Mode in FIPS Mode

Protocol	Key Exchange	Auth	Cipher	Integrity
SSHv2[6]	Diffie-Hellman (L = 2048, N=2047) EC Diffie-Hellman P-256[7] EC Diffie-Hellman P-384[8] EC Diffie-Hellman P-521[9]	ECDSA P-256 ECDSA P-384 ECDSA P-521 RSA 2048 RSA 3072 RSA 4096	Triple-DES CBC[10] AES CBC 128/192/256 AES CTR 128/192/256	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-512

No part of these protocols, other than the KDF, have been tested by the CAVP and CMVP. The SSH protocol allows independent selection of key exchange, authentication, cipher and integrity. In Table 9 above, each column of options for a given protocol is independent and may be used in any viable combination. These security functions are also available in the SSH connect (non-compliant) service

2.4 Disallowed Algorithms

These algorithms are non-Approved algorithms that are disabled when the module is operated in an Approved mode of operation.

- ARCFOUR
- Blowfish
- CAST

[5] ECDH is only allowed until June 30, 2022.

[6] RFC 4253 governs the generation of the Triple-DES encryption key for use with the SSHv2 protocol

[7] ECDH is only allowed until June 30, 2022.

[8] ECDH is only allowed until June 30, 2022.

[9] ECDH is only allowed until June 30, 2022.

[10] Use of Triple-DES in this module is only allowed until December 31, 2023.

- DSA (SigGen, SigVer; non-compliant)
- HMAC-MD5
- HMAC-RIPEMD160
- UMAC

2.5 Critical Security Parameters

All CSPs and public keys used by the module are described in this section.

Table 10 – Critical Security Parameters (CSPs)

Name	Description and usage
DRBG Seed	Seed material used to seed or reseed the HMAC DRBG
DRBG State	V and Key values for the HMAC DRBG
DRBG Entropy Input	256 bits entropy (min) input used to instantiate HMAC DRBG
SSH PHK	SSH Private host key. 1st time SSH is configured, the keys are generated. ECDSA P-256 by default, but also supports ECDSA P-384, ECDSA P-521, RSA 2048 and RSA 3072. Used to identify the host.
SSH DH/ECDH[11]	SSH Diffie-Hellman private component. Ephemeral Diffie-Hellman private key used in SSH. DH (L=2048, N=2047)[12], ECDH P-256, ECDH P-384 or ECDH P-521
SSH-SEKs	SSH Session Keys (derived using SP 800-135 KDF): SSH Session Encryption Key: Triple-DES (3key) or AES; SSH Session Integrity Key: HMAC.
HMAC key	The LibMD HMAC keys: message digest for hashing password and critical function test.
CO-PW	Password used to authenticate the CO. Password is input as plaintext via serial port or encrypted via SSH.
User-PW	Password used to authenticate the User. Password is input as plaintext via serial port or encrypted via SSH.

Table 11 – Public Keys

Name	Description and usage
SSH-PUB	SSH Public Host Key used to identify the host. ECDSA P-256 by default, but also supports ECDSA P-384, ECDSA P-521, RSA 2048 and RSA 3072.
SSH-DH-PUB[13]	Diffie-Hellman public component. Ephemeral Diffie-Hellman public key used in SSH key establishment. DH (L=2048, N=2047), ECDH P-256, ECDH P-384 or ECDH P-521
Auth-User Pub	User Authentication Public Keys. Used to authenticate users to the module. ECDSA P-256, P-384, P-521, RSA 2048, RSA 3072
Auth-CO Pub	CO Authentication Public Keys. Used to authenticate CO to the module. ECDSA P-256, P-384, P-521, RSA 2048, RSA 3072 or RSA 4096

[11] ECDH is only allowed until June 30, 2022.

[12] SSH generates a Diffie-Hellman private key that is 2x the bit length of the longest symmetric or MAC key negotiated.

[13] ECDH is only allowed until June 30, 2022.

Root-CA	ECDSA P-256 X.509 Certificate; Used to verify the validity of the Juniper Package CA at software load and also at runtime for integrity.
Package-CA	ECDSA P-256 X.509 Certificate; Used to verify the validity of Juniper images at software load and boot.

3 Roles, Authentication and Services

3.1 Roles and Authentication of Operators to Roles

The module supports two roles: Cryptographic Officer (CO) and User. The module supports concurrent operators, but does not support a maintenance role and/or bypass capability. The module enforces the separation of roles using either of the identity-based operator authentication methods in Section 3.2.

The Cryptographic Officer role configures and monitors the module via a console or SSH connection. As root or super-user, the Cryptographic Officer has permission to view and edit secrets within the module.

The User role monitors the switch via the console or SSH. The user role cannot not change the configuration.

3.2 Authentication Methods

The module implements two forms of Identity-based authentication: username and password over the Console and SSH, as well as username and public key over SSH.

Password authentication

The module enforces 10-character passwords (at minimum) chosen from the 96 human readable ASCII characters. The maximum password length is 20-characters; thus the probability of a successful random attempt is $1/96^{10}$, which is less than 1/1,000,000.

The module enforces a timed access mechanism as follows: For the first two failed attempts (assuming 0 time to process), no timed access is enforced. Upon the third attempt, the module enforces a 5-second delay. Each failed attempt thereafter results in an additional 5-second delay above the previous (e.g. 4th failed attempt = 10-second delay, 5th failed attempt = 15-second delay, 6th failed attempt = 20-second delay, 7th failed attempt = 25-second delay).

This leads to a maximum of 9 possible attempts in a one-minute period for each *getty*. The best approach for the attacker would be to disconnect after 4 failed attempts and wait for a new *getty* to be spawned. This would allow the attacker to perform roughly 9.6 attempts per minute (576 attempts per hour/60 mins); this would be rounded down to 9 per minute, because there is no such thing as 0.6 attempts. Thus the probability of a successful random attempt is $1/96^{10}$, which is less than 1/1 million. The probability of a success with multiple consecutive attempts in a one-minute period is $9/(96^{10})$, which is less than 1/100,000.

Signature verification

Public key authentication in SSH uses either RSA (2048 and 3072 bit moduli) or ECDSA signature (P-256, P-384 and P-521). Let x denote the maximum number of signature verifications that the IUT can perform in a minute. Assuming a minimum security strength of 112 bits (corresponding to RSA with 2048-bit moduli as per SP800-57 Part1 Rev3), the probability of a successful brute-force attack with multiple consecutive attempts in a one-minute period is $x/2^{112}$. For this probability to be greater than 1/100,000, the number of verifications per minute must be $x > \frac{2^{112}}{10^5} \cong 2^{197}$, which is clearly an infeasible amount of signature verifications. If the IUT were able to compute one signature verification per CPU cycle, this would amount to $60 \times 4 \times 2.2 \times 10^9 \cong 2^{39}$ verifications per minute for the 2.2 GHz quad-core Intel CPU shared by all IUT models.

3.3 Services

All services implemented by the module are listed in the tables below. Table 14 lists the access to CSPs by each service.

Table 12 – Authenticated Services

Service	Description	CO	User
Configure security	Security relevant configuration	X	
Configure	Non-security relevant configuration	X	
Status	Show status	X	X
Zeroize	Destroy all CSPs	X	
SSH connect	Initiate SSH connection for SSH monitoring and control (CLI)	X	X
Console access	Console monitoring and control (CLI)	X	X
Remote reset	Software initiated reset	X	
Load image	Verification and loading of a validated firmware image into the switch.	X	

Table 13 – Unauthenticated services

Service	Description
Local reset	Hardware reset or power cycle
Traffic	Traffic requiring no cryptographic services

Table 14 – CSP Access Rights within Services

SERVICE	CSPs								
	DRBG Seed	DRBG State	DRBG Entropy Input	SSH PHK	SSH DH/ECDH	SSH-SEK	HMAC Key	CO-PW	User-PW
Configure security	--	E	--	GWR	--	--	G	W	W
Configure	--	--	--	--	--	--	--	--	--
Status	--	--	--	--	--	--	--	--	--
Zeroize	Z	Z	Z	Z	Z	Z	--	Z	Z
SSH connect	--	E	--	E	GE	GE	--	E	E
Console access	--	--	--	--	--	--	--	E	E
Remote reset	GEZ	GZ	GZ	--	Z	Z	Z	Z	Z
Local reset	GEZ	GZ	GZ	--	Z	Z	--	Z	Z
Traffic	--	--	--	--	--	--	--	--	--
Load Image	--	--	--	--	--	--	--	--	--

G = Generate: The module generates the CSP

R = Read: The CSP is read from the module (e.g. the CSP is output)

E = Execute: The module executes using the CSP

W = Write: The CSP is updated or written to the module

Z = Zeroize: The module zeroizes the CSP.

Table 15 – Public Key Access Rights within Services

	SSH-PUB	SSH-DH-PUB	Auth-User Pub	Auth-CO Pub	Root-CA	Package-CA
Configure security	GWR	--	W	W	--	--
Configure	--	--	--	--	--	--
Status	--	--	--	--	--	--
Zeroize	Z	--	Z	Z	--	--
SSH connect	E	GE	E	E	--	--
Console access	--	--	--	--	--	--
Remote reset	--	Z	Z	Z	--	E
Local reset	--	Z	Z	Z	--	E
Traffic	--	--	--	--	--	--
Load Image	--	--	--	--	EW	EW

G = Generate: The module generates the CSP

R = Read: The CSP is read from the module (e.g. the CSP is output)

E = Execute: The module executes using the CSP

W = Write: The CSP is updated or written to the module

Z = Zeroize: The module zeroizes the CSP.

3.4 Non-Approved Services

The following services are available in the non-Approved mode of operation. The security functions provided by the non-Approved services are identical to the Approved counterparts except for SSH Connect (non-compliant). SSH Connect (non-compliant) supports the security functions identified in Section 2.4 and Table 9.

Table 16 – Authenticated Services

Service	Description	CO	User
Configure security (non-compliant)	Security relevant configuration	X	
Configure (non-compliant)	Non-security relevant configuration	X	
Status (non-compliant)	Show status	X	X
Zeroize (non-compliant)	Destroy all CSPs	X	
SSH connect (non-compliant)	Initiate SSH connection for SSH monitoring and control (CLI)	X	X
Console access (non-compliant)	Console monitoring and control (CLI)	X	X
Remote reset (non-compliant)	Software initiated reset	X	

Table 17 – Unauthenticated traffic

Service	Description
Local reset (non-compliant)	Hardware reset or power cycle
Traffic (non-compliant)	Traffic requiring no cryptographic services

4 Self-tests

Each time the module is powered up, it tests that the cryptographic algorithms still operate correctly and that sensitive data have not been damaged. Power-up self–tests are available on demand by power cycling the module.

On power up or reset, the module performs the self-tests described below. All KATs must be completed successfully prior to any other use of cryptography by the module. If one of the KATs fails, the module enters the Critical Failure error state.

The module performs the following power-up self-tests:

- Firmware Integrity check using ECDSA P-256 with SHA-256
- Kernel KATs
  - HMAC-SHA-1 KAT
  - HMAC-SHA-256 KAT
  - SHA-384 KAT
  - SHA-512 KAT
  - SP 800-90A HMAC DRBG KAT
    - Health-tests initialize, re-seed, and generate.
- OpenSSL KATs
  - ECDSA P-256 Sign/Verify PCT
  - ECDH P-256 KAT
    - Derivation of the expected shared secret.
  - DH (L=2048, N=256) KAT *Derivation of the expected shared secret
  - RSA 2048 w/ SHA-256 Sign KAT
  - RSA 2048 w/ SHA-256 Verify KAT

  - Triple-DES-CBC Encrypt KAT
  - Triple-DES-CBC Decrypt KAT
  - HMAC-SHA-1 KAT
  - HMAC-SHA-256 KAT
  - HMAC-SHA-384 KAT
  - HMAC-SHA-512 KAT
  - AES-CBC (128/192/256) Encrypt KAT
  - AES-CBC (128/192/256) Decrypt KAT
  - KDF-SSH KAT
  - SP 800-90A HMAC DRBG KAT
    - Health-tests initialize, re-seed, and generate.
- LibMD KATs
  - HMAC SHA-1
  - HMAC SHA-256
  - SHA-512

The module also performs the following conditional self-tests:

- Continuous RNG Test on the SP 800-90A HMAC-DRBG
- Continuous RNG test on the NDRNG
- Pairwise consistency test when generating ECDSA, and RSA key pairs
- SP800-56A assurances as per SP 800-56A Sections 5.5.2,5.6.2, and/or 5.6.3, in accordance to IG 9.6.
- Firmware Load Test (ECDSA P-256 with SHA-256 signature verification)

5 Security Rules and Guidance

The module design corresponds to the security rules below. The term *must* in this context specifically refers to a requirement for correct usage of the module in the Approved mode; all other statements indicate a security rule implemented by the module.

1. The module clears previous authentications on power cycle.
2. When the module has not been placed in a valid role, the operator does not have access to any cryptographic services.
3. Power up self-tests do not require any operator action.
4. Data output is inhibited during key generation, self-tests, zeroization, and error states.
5. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
6. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
7. The module does not support a maintenance interface or role.
8. The module does not support manual key entry.
9. The module does not output intermediate key values.
10. The module requires two independent internal actions to be performed prior to outputting plaintext CSPs.
11. The cryptographic officer must determine whether firmware being loaded is a legacy use of the firmware load service (legacy being those Junos firmware images signed with RSA signatures instead of ECDSA).
12. The cryptographic officer must retain control of the module while zeroization is in process.
13. Use of SSH ECDH is only allowed until June 30th, 2022. From July 1st, 2022, the module should be configured to disallow the use of ECDH in SSH.

6 References and Definitions

The following standards are referred to in this Security Policy.

Table 18 – References

Abbreviation	Full Specification Name
[FIPS140-2]	Security Requirements for Cryptographic Modules, May 25, 2001
[SP800-131A]	Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, Revision 1, March 2019
[IG]	Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program
[135]	National Institute of Standards and Technology, Recommendation for Existing Application-Specific Key Derivation Functions, Special Publication 800-135rev1, December 2011.
[186-4]	National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4, July, 2013.
[197]	National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, November 26, 2001
[38A]	National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, Special Publication 800-38A, December 2001
[38D]	National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, Special Publication 800-38D, November 2007
[56A]	National Institute of Standards and Technology, Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography, Special Publication 800-56A, March 2007
[56ARev3]	National Institute of Standards and Technology, Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography, Special Publication 800-56A Revision 3, April 2018
[198]	National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198-1, July, 2008
[180]	National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standards Publication 180-4, August, 2015
[67]	National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Special Publication 800-67, Revision 2, November 2017
[90A]	National Institute of Standards and Technology, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Special Publication 800-90A, June 2015.

Abbreviation	Full Specification Name
[133]	National Institute of Standards and Technology, Recommendation for Cryptographic Key Generation, Special Publication 800-133, Revision 1, July 2019

Table 19 – Acronyms and Definitions

Acronym	Definition
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EMI/EMC	Electromagnetic Interference/Electromagnetic Compatibility
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Message Authentication Code
IKE	Internet Key Exchange Protocol
IPsec	Internet Protocol Security
MD5	Message Digest 5
RSA	Public-key encryption technology developed by RSA Data Security, Inc.
SHA	Secure Hash Algorithms
SSH	Secure Shell
Triple-DES	Triple - Data Encryption Standard

Table 20 – Datasheets

Model	Title	URL
EX4650	EX4650 Ethernet Switch	https://www.juniper.net/assets/us/en/local/pdf/datasheets/1000640-en.pdf
QFX5120	QFX5120 Ethernet Switch	https://www.juniper.net/assets/us/en/local/pdf/datasheets/1000639-en.pdf
QFX5210	QFX5210 Switch	https://www.juniper.net/assets/us/en/local/pdf/datasheets/1000633-en.pdf