QFX5120-48YM switch on JUNOS 23.4R2 Security Target

Document Version: 1.0

1133 Innovation Way
Sunnyvale, CA 94089
USA

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History

Version	Date	Changes
Version 0.1	February 02, 2025	Initial Release
Version 0.2	March 03, 2025	Addressed the lead review comments
Version 0.3	April 17, 2025	Vendor provided responses to the TSS related queries (via Comments)
Version 0.4	May 1, 2025	Incorporated vendor's responses for the TSS section
Version 0.5	May 26, 2025	Added processor
Version 0.6	June 24, 2025	Finalized SFR selections and addressed EOR
Version 0.7	September 29, 2025	Updated CAVP table
Version 0.8	October 13, 2025	Updated for Check-in ECR comments
Version 0.9	January 14, 2026	Updated for 2nd round Check-in ECR comments
Version 1.0	March 27, 2026	Updated for check-out ECR comments

Contents

1. INTRODUCTION

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE, and the functional and assurance requirements that the TOE meets.

1.1 SECURITY TARGET AND TOE REFERENCE

This section provides the information needed to identify and control the TOE and the ST.

Table 1 – TOE/ST Identification

Category	Identifier
ST Title	QFX5120-48YM switch on JUNOS 23.4R2 Security Target
ST Version	1.0
ST Date	March 27, 2026
ST Author	Intertek Acumen Security
TOE Identifier	QFX5120-48YM switch on JUNOS 23.4R2
TOE Version	23.4R2
TOE Developer	HPE Juniper Networking
Key Words	Network Device

1.2 TOE OVERVIEW

The TOE is the HPE Juniper Networking QFX5120-48YM switch on JUNOS 23.4R2 with MACsec.

The following appliance model constitutes the TOE:

- QFX5120 with an Intel Xeon D-1627 (Hewitt Lake-DE) running Junos OS 23.4R2 software, offering 48 25GbE (SFP28)/10GbE (SFP+)/1GbE (SFP) ports.
- AES ECB 128bit & 256bit Encryption/Decryption Engine (BCM82391)

The TOE is a secure network device that protects itself largely by offering only a minimal logical interface to the network and attached nodes. It includes the Junos OS firmware, JUNOS OS 23.4R2, which is a special purpose OS offering no general-purpose computing capabilities. Junos OS implements both management and control functions as well as all IP routing.

The appliance's primary function is to support the definition and enforcement of information flow policies among network nodes. Each information flow from one network node to other passes through an instance of the TOE. Information flow is controlled based on network node addresses and protocol. The TOE also ensures that security-relevant activity is audited and implements the necessary tools to manage all security functions.

The TOE implements a variety of high-speed interfaces (only Ethernet is in the scope of the evaluation) for enterprise branch, campus, and data center networks. The appliance is physically self-contained, housing the firmware and hardware necessary to perform all routing functions. The architecture components of the TOE are:

- Switch fabric – the switch fabric boards/modules provide a highly scalable, non-blocking, centralized switch fabric matrix through which all network data passes.
- Routing Engine (Control Board) – the Routing Engine (RE) runs the Junos firmware and implements Layer 3 routing services and Layer 2 switching services. The RE also implements the management functions for configuration and operation of the TOE and controls the flow of information through the TOE, including support for appliance interface control and control plane functions such as chassis component, system management and user access to the appliance.
- Layer 2 switching services, Layer 3 switching/routing services and network management for all operations necessary for the configuration and operation of the TOE and controls the flow of information through the TOE.
- Packet Forwarding Engine (PFE) – The PFE implements all operations necessary for transit packet forwarding. The PFE implements an extensive set of Layer 2 and Layer 3 services that can be deployed in any combination of L2- L3 applications.
- Power – The TOE includes non-PoE ports. Power supply bays allow flexibility for provisioning and redundancy. The power supplies connect to the midplane, which distributes the different output voltages produced by the power supplies to the appliance components, depending on their voltage requirements.

The appliance supports numerous routing and switching standards for flexibility and scalability. Juniper's Virtual Chassis technology allows multiple interconnected switches to operate as a single, logical unit, enabling users to manage all platforms as one virtual device. The functions of the appliances can all be managed through the Junos firmware, either from a connected terminal console or via a network connection. Network management is secured using the SSH protocol. All management, whether from a user connecting to a terminal or from the network, requires successful authentication. In the evaluated deployment the TOE is managed and configured via Command Line Interface, either via a directly connected console or over the network secured using the SSH protocol.

The TOE implements MACsec between adjacent devices. All traffic communicated between the devices including frames for LLDP (Link Layer Discovery Protocol), DHCP (Dynamic Host Configuration Protocol), ARP (Address Resolution Protocol), STP (Spanning Tree Protocol), Ethernet Control frames, etc. (the exceptions to this protection are Destination MAC and Source MAC addresses in MACsec and MKA frames).

MACsec can be deployed in point-to-point mode or shared mode with multiple stations. In the evaluated configuration MACsec must be configured individually on each point-to-point Ethernet link, such that a pair of MACsec devices (connected by a physical medium) protect Ethernet frames switched or routed from one device to the other. The two MACsec devices are provided with a Connectivity Association Key (CAK) and utilize the MACsec Key Agreement (MKA) protocol to create a secure tunnel. MKA is used by the two MACsec devices to agree upon MACsec keys. MACsec must be configured to protect all traffic between the devices, with the exception of the MKA or Ethernet control traffic such as EAP over LAN (EAPOL) frames. The devices will first exchange MKA frames, which serve to determine if the peer is an authorized peer and agree upon a shared key and MACsec cipher suite used to set up a transmit (Tx) Security Association (SA) and a receive (Rx) SA. Once the SAs are set up, MACsec-protected frames traverse the unprotected link.

1.3 TOE DESCRIPTION

This section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references.

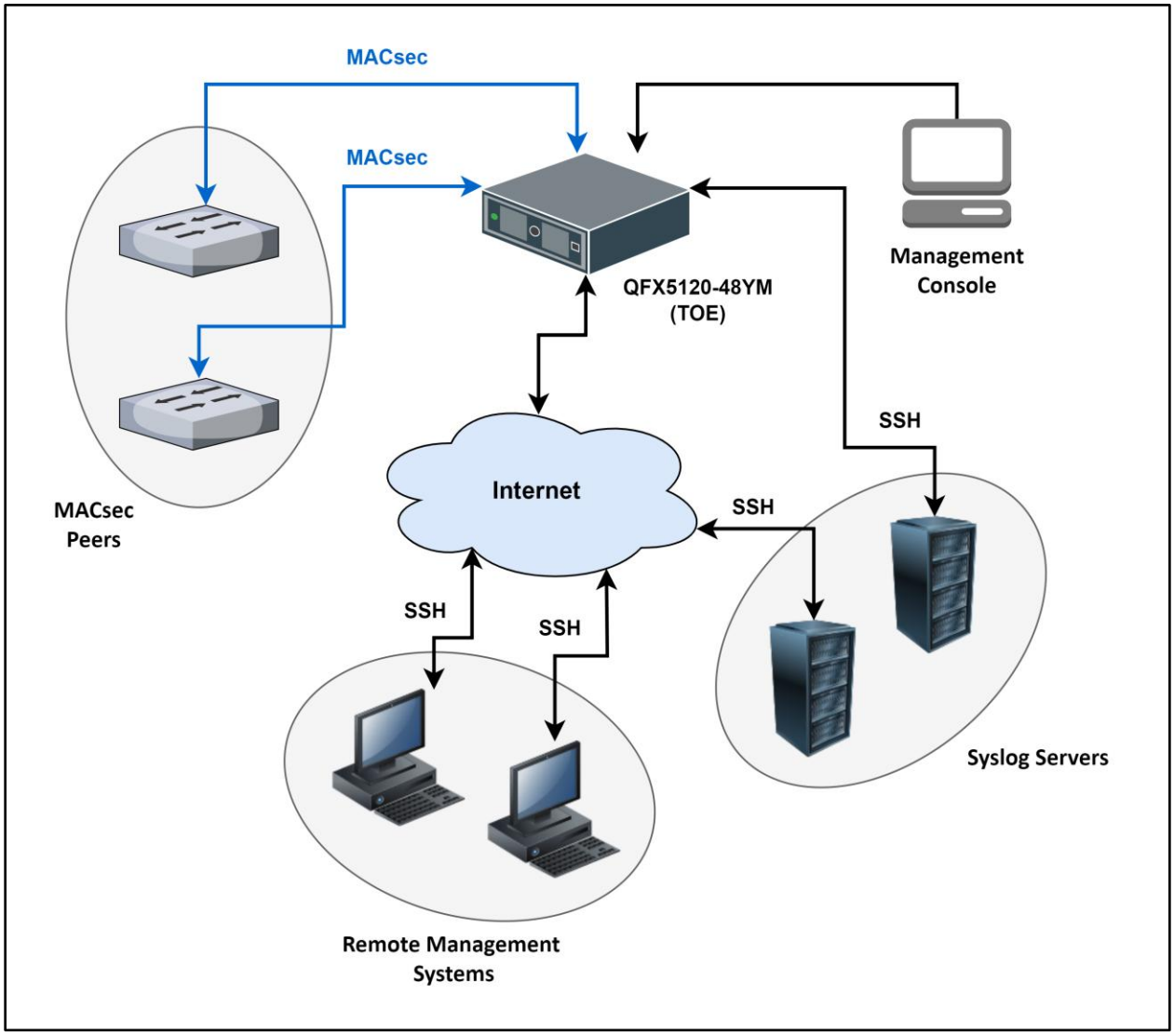

Figure 1 – Representative TOE Deployment

1.3.1 PHYSICAL BOUNDARIES

The TOE is a hardware appliance (network switch) which is comprised of hardware components running on JUNOS-OS software. The boundaries are illustrated in Figure 2 – TOE Boundaries.

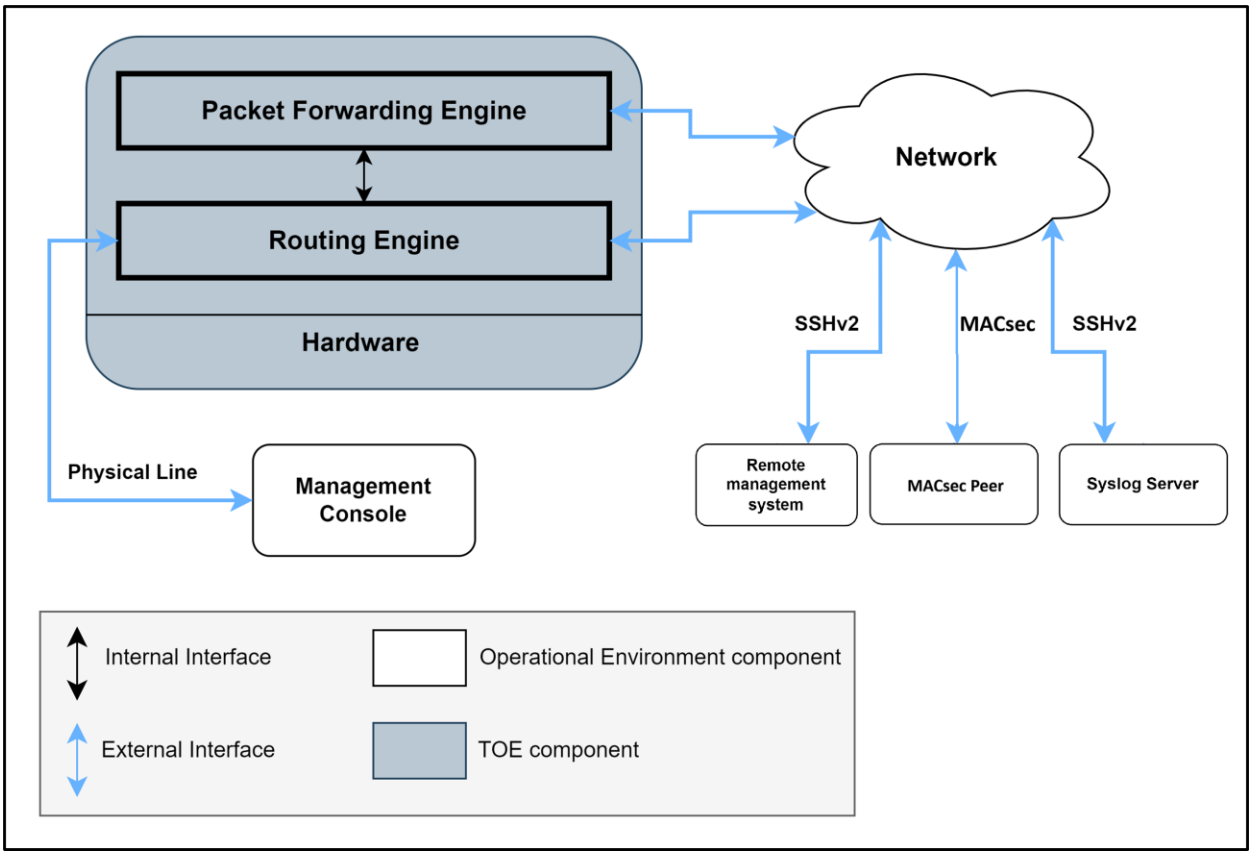

Figure 2 – TOE Boundaries

The physical boundary of the TOE is the entire chassis of the appliance. The TOE interfaces are comprised of the network interfaces which pass traffic, and the management interface which handle administrative actions.

Table 2 – TOE Physical Boundary Components

Component	Required	Purpose/Description
Hardware – Switch fabric	Yes	The switch fabric boards/modules provide a highly scalable, non-blocking, centralized switch fabric matrix through which all network data passes.
Routing Engine (Control Board)	Yes	The Routing Engine (RE) runs the Junos firmware and implements Layer 3 routing services and Layer 2 switching services. The RE also implements the management functions for configuration and operation of the TOE and controls the flow of information through the TOE, including support for appliance interface control and control plane functions such as chassis component, system management and user access to the appliance.
Packet Forwarding Engine (PFE)	Yes	The PFE implements all operations necessary for transit packet forwarding. The PFE implements an extensive set of Layer 2

Component	Required	Purpose/Description
		and Layer 3 services that can be deployed in any combination of L2- L3 applications.

The RE and PFE perform their primary tasks independently, while constantly communicating through a high-speed internal link. This arrangement provides streamlined forwarding and routing control and the capability to run Internet-scale networks at high speeds.

1.3.2 SECURITY FUNCTIONS PROVIDED BY THE TOE

The TOE provides the security functions required by the Collaborative Protection Profile for Network Devices, hereafter referred to as 'NDcPP v3.0e' or 'NDcPP', PP-Module for MACsec Ethernet Encryption, hereafter referred to as 'MOD_MACsec V1.0' and Functional Package for SSH Version 1.0, hereafter referred to as 'PKG_SSH_v1.0'.

1.3.2.1 SECURITY AUDIT

The TOE generates audit events for all start-up and shutdown functions as well as all auditable events specified in Table 12 – Security Functional Requirements and Auditable Events and Table 13 – Security Functional Requirements and Auditable Events/MACsec. Auditable events are stored in the syslog files on the appliance and can be sent to an external log server (via Netconf over SSH). Auditable events include start-up and shutdown of the audit functions, authentication events, and all SFR-specific events required by the applicable Protection Profiles. Audit records include the date and time, event category, event type, username, and the outcome of the event (success or failure). Local syslog storage limits are configurable and monitored. If the storage limit is reached the oldest logs will be overwritten.

1.3.2.2 CRYPTOGRAPHIC SUPPORT

The TOE implements an SSH server to support protected communications for administrators to establish secure sessions and to connect to external syslog servers. The TOE requires that applications exchanging information with it are successfully authenticated prior to any exchange (i.e. applications connecting over SSH). Communication over point-to-point links between Juniper appliances can be secured using MACsec. The TOE includes cryptographic modules that implement the underlying cryptographic services, including key management and protection of stored keys, algorithms, random bit generation and crypto administration. The cryptographic modules provide confidentiality and integrity services for authentication and for protecting communications with connecting applications. Details on cryptographic implementations by the TOE are mentioned in the Table 3 – TOE Cryptography Implementation.

Table 3 – TOE Cryptography Implementation

Cryptographic Methods	Usage
FCS_CKM.1 Cryptographic Key Generation	• Cryptographic key generation conforming to FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1; ○ RSA key sizes supported are: 2048 bits, 3072 bits, and 4096 bits.

Cryptographic Methods	Usage
	• Cryptographic key generation conforming to FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Appendix A.2. ○ Elliptic NIST curves supported are: P-256, P-384, and P-521.
FCS_CKM.2 Cryptographic Key Establishment	• Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”.
FCS_CKM.4 Cryptographic Key Destruction	• Refer to Table 18 – Zeroization of Keys and CSP for Key Zeroization details.
FCS_COP.1/DataEncryption Cryptographic Operations (AES Data Encryption/Decryption)	• AES encryption and decryption conform with ISO 18033-3. • AES key size supported is 128 and 256 bits. • AES mode supported is CBC, CTR and GCM as specified in ISO 10116, ISO 10116 and ISO 19772 respectively.

FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)	• RSA digital signature algorithm conforming to FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5.
	  ○ RSA key sizes supported are: 2048 bits, 3072 bits, and 4096 bits.
	• Elliptical curve digital signature algorithm conforming to FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves.
	  ○ Elliptic NIST curves supported are: P-256, P-384, and P-521.

FCS_COP.1/Hash Cryptographic Operations (Hash Algorithm)	• Cryptographic hashing services conforming to ISO/IEC 10118-3:2004. • Hashing algorithms supported are SHA-1, SHA-256, SHA-384, and SHA-512. • Message digest sizes supported are: 160, 256, 384, and 512 bits.
FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)	• Keyed-hash message authentication conforming to ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm2”. • Keyed hash algorithm supported are: HMAC-SHA-256, and HMAC-SHA512. • Key sizes supported are: 256 and 512 bits. • Message digest sizes supported are: 256, and 512 bits.

Cryptographic Methods	Usage
FCS_COP.1/CMAC Cryptographic Operation (AES-CMAC Keyed Hash Algorithm)	• Keyed-hash message authentication as per NIST SP 800-38B standard • Algorithm: AES-CMAC • Key Sizes: 128, 256 bits • Digest Size: 128 bits

FCS_COP.1/MACSEC Cryptographic Operation (MACsec AES Data Encryption/Decryption)	• Key Wrap:
	  ○ Algorithm: AES Key Wrap
	  ○ Standard: NIST SP 800-38F
	  ○ Key Sizes: 128, 256 bits
	• Data Encryption/Decryption:
	  ○ Algorithm: AES in Galois/Counter Mode (GCM)
	  ○ Standard: ISO 19772
	  ○ Key Sizes: 128, 256 bits

FCS_RBG_EXT.1 Random Bit Generation	• Deterministic Random Bit Generation:
	  ○ Standard: ISO/IEC 18031:2011
	  ○ Algorithm: HMAC_DRBG
	  ○ Hash Functions: SHA-512
	• Entropy Source:
	  ○ Type: Software-based noise source
	  ○ Minimum Entropy: 256 bits
	  ○ Entropy Strength: At least equal to the greatest security strength of generated keys and hashes (as per ISO/IEC 18031:2011 Table C.1)

FCS_SSHS_EXT.1 SSH Server Protocol and FCS_SSH_EXT.1 SSH Protocol	• The TOE supports SSH v2 protocol compliant to the following RFCs:4251, 4252, 4253, 4254, 4256, 4344, 5656, 6668, 8308 and 8332. • The TOE supports password-based and public-key-based authentication. • SSH public-key authentication uses ssh-rsa (RFC 4253), rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC 5656), ecdsa-sha2-nistp384 (RFC 5656), ecdsa-sha2-nistp521 (RFC 5656), • SSH transport uses the following encryption algorithms: aes128-ctr (RFC 4344), aes256-ctr (RFC 4344), aes128-cbc (RFC 4253), aes256-cbc (RFC 4253). • Packets greater than 256K bytes in an SSH transport connection are dropped. • SSH transport uses the following data integrity MAC algorithms: hmac-sha2-256 (RFC 6668), and hmac-sha2-512 (RFC 6668).

Cryptographic Methods	Usage
	• Key exchange algorithms supported are: ecdh-sha2-nistp256 (RFC 5656), ecdh-sha2-nistp384 (RFC 5656) and ecdh-sha2-nistp521 (RFC 5656). • The TOE ensures that during SSH connections, the same session keys are used for a threshold of no longer than one hour and no more than one gigabyte of transmitted data.

FCS_MACSEC_EXT.1 MACsec	• MACsec Implementation:
	  ○ Standard: IEEE 802.1AE-2018
	• Secure Channel Identifier (SCI):
	  ○ Derivation: From peer's MAC address and port
	  ○ Validation: Reject MPDUs with incorrect SCI during a session
	• Permitted EtherTypes:
	  ○ EAPOL (88-8E)
	  ○ MACsec frames (88-E5)
	  ○ MAC control frames (88-08)

FCS_MACSEC_EXT.2 MACsec Integrity and Confidentiality	• MACsec Implementation:
	  ○ Integrity Protection: Supported
	  ○ Confidentiality Offset: 0, 30, 50
	• Integrity Check Value (ICV):
	  ○ Derivation: Using the Secure Association Key (SAK)
	  ○ Size: 8-16 octets
	  ○ Protection: Destination/Source MAC addresses and all MPDU fields
	• Integrity Check Value Key (ICK):
	  ○ Derivation: From Connectivity Association Key (CAK) using a Key Derivation Function (KDF)

FCS_MACSEC_EXT.3 MACsec Randomness	• Secure Association Key (SAK) Generation:
	  ○ Method: Key derivation from Connectivity Association Key (CAK) per IEEE 802.1X-2020 section 9.8.1
	  ○ Uniqueness Probability: No less than 1 in 2 powers of the generated key size
	• SAK Nonce Generation:
	  ○ Source: TOE's random bit generator (as specified by FCS_RBG_EXT.1)

Cryptographic Methods	Usage

FCS_MACSEC_EXT.4 MACsec Key Usage	• Peer Authentication:
	  ○ Method: Pre-shared keys (PSKs) [no other method]
	• SAK Distribution:
	  ○ Method: AES key wrap
	• CAK Lifetime:
	  ○ Support: Specifiable lifetime
	• Connectivity Association Key Names (CKNs):
	  ○ Association: With SAKs, defined by KDF using CAK as input data (per IEEE 802.1X-2020, Section 9.8.1)
	• CKN Association: With CAKs
	  ○ Length: 1-32 octets

FCS_MKA_EXT.1 MACsec Key Agreement	• Key Agreement Protocol (MKA):
	  ○ Standard: IEEE 802.1X-2020 and 802.1Xbx-2014
	• MKPDU Integrity:
	  ○ Protection: Using ICV derived from ICK
	  ○ ICK Derivation: From CAK using a KDF
	• MKA Timeouts:
	  ○ Lifetime Timeout: 6.0 seconds
	  ○ Bounded Hello Timeout: 2 seconds
	• SAK Refresh/Distribution (Key Server):
	  ○ Refresh: On expiration
	  ○ Distribution: By pre-shared key (PSK) pairwise CAKs
	  ○ Distribution: Fresh SAK on member addition/removal
	• MKPDU Validation:
	  ○ Standard: IEEE 802.1X-2020 Section 11.11.2
	  ○ Discard Conditions:
	  ▪ Individual destination address
	  ▪ Length < 32 octets
	  ▪ Length < (Basic Parameter Set body length + 16 octets ICV)
	  ▪ Unrecognized CAK Name
	• MKPDU Processing (Post-Validation):
	  ○ Algorithm Agility Parameter:

Cryptographic Methods	Usage
	▪ Implemented Algorithm: ICV verification per IEEE 802.1X-2020 Section 9.4.1 ▪ Unrecognized/Unimplemented Algorithm: Discard MKPDU ○ Decoding: Per IEEE 802.1X-2020 Section 11.11.4 (for validated and verified MKPDUs)

1.3.2.3 IDENTIFICATION AND AUTHENTICATION

TOE supports Role Based Access Control. All users must be authenticated to the TOE prior to being granted access to any management actions. The TOE supports password-based authentication and public key-based authentication. Based on the assigned role, a user is granted a set of privileges to access the system. Administrative users must provide unique identification and authentication data before any administrative access to the system is granted. Authentication data entered and stored on the TOE is protected.

1.3.2.4 SECURITY MANAGEMENT

The TOE provides a Security Administrator role that is responsible for

- configuration and maintenance of cryptographic elements related to the establishment of secure connections to and from the evaluated product;
- regular review of all audit data;
- initiation of trusted update function;
- administration of MACsec functionality;
- all administrative tasks (e.g., creating the security policy).

The devices are managed through a Command Line Interface (CLI). The CLI is accessible through local (serial) console connection or remote administrative (SSH) session.

1.3.2.5 PROTECTION OF THE TSF

The TOE protects all passwords, pre-shared keys, symmetric keys and private keys from unauthorized disclosure. Passwords are stored using sha256 or sha512. The TOE executes self-tests during initial start-up to ensure correct operation and enforcement of its security functions. An administrator can install software updates to TOE. The TOE internally maintains the date and time.

1.3.2.6 TOE ACCESS

The TOE displays a customizable banner before any administrative session can be established with it. The TOE will terminate local or remote interactive sessions after a specified period of session inactivity configured by an administrator. An administrator can terminate their own interactive local or remote sessions.

1.3.2.7 TRUSTED PATH/CHANNELS

The TOE supports SSH for secure communications with authorized IT entities such as syslog servers. The TOE supports SSHv2 (remote CLI) for secure remote administration. The TOE also supports MACsec for securing data at Layer 2 between TOE and MACsec supporting peer device.

1.3.3 TOE DOCUMENTATION

The following documents are essential to understanding and controlling the TOE in the evaluated configuration:

- Common Criteria Evaluated Configuration Guide for QFX5120-48YM Device, Release 23.4R2.

1.3.4 REFERENCES

In addition to TOE documentation, the following reference may also be valuable when understanding and controlling the TOE:

The PP-config for this evaluation is CFG_NDcPP-MACsec_v2.0 which includes the following PP/MOD:

- collaborative Protection Profile for Network Devices, Version 3.0e (CPP_ND_V3.0E)
- PP-Module for MACsec Ethernet Encryption, Version 1.0 (MOD_MACsec_V1.0)

Also, additional Packages are claimed as follows:

- Functional Package for SSH Version 1.0 conformant (PKG_SSH_v1.0)

1.4 TOE ENVIRONMENT

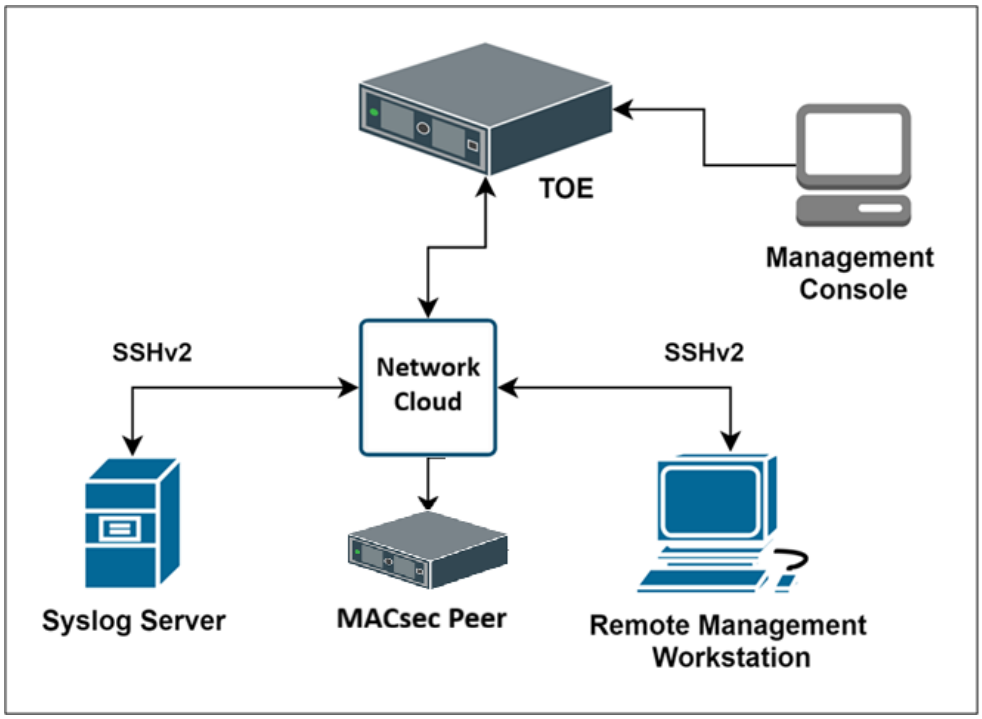

Figure 3 – TOE Environment

The following environmental components are required to operate the TOE in the evaluated configuration:

Table 4 – Required Environmental Components

Components	Description
Local Management Console	Any management workstation (computer) that is directly (serially) connected to the TOE's console port may be used by the TOE administrator for local administration of the TOE.
Remote Management Workstation	Any management workstation (computer) with a SSHv2 client installed that may be used by the TOE administrator for remote administration of the TOE through SSH protected channel.
Syslog Server	A syslog server, used for remote storage of audit records that have been generated by and transmitted from the TOE securely using SSH tunnel (SSHv2).
MACsec Peer	A MACsec Peer, used for establishing MACsec communication for securing Layer 2 data using MACsec functionality.

1.5 PRODUCT FUNCTIONALITY NOT INCLUDED IN THE SCOPE OF THE EVALUATION

The following product functionality is not included in the CC evaluation:

- Use of telnet, since it violates the Trusted Path requirement set.
- Use of FTP, since it violates the Trusted Path requirement set.
- Use of SNMP, since it violates the Trusted Path requirement set.
- Use of SSL, including management via J-Web, JUNOScript and JUNOScope, since it violates the Trusted Path requirement set.
- Use of CLI account super-user and linux root account.
- Use of NTP server to obtain timestamp data.

2. CONFORMANCE CLAIMS

This section identifies the TOE conformance claims, conformance rationale, and relevant Technical Decisions (TDs).

2.1 CC CONFORMANCE CLAIMS

The TOE is conformant to the following:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017 (Extended)
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017 (Conformant)

2.2 PROTECTION PROFILE CONFORMANCE

This ST claims exact conformance to the following:

- PP-Configuration for Network Devices and MACsec Ethernet Encryption, Version 2.0, 2023-03-29 [CFG_NDcPP-MACsec_v2.0].
  This PP-Configuration includes the following:
  - collaborative Protection Profile for Network Devices, Version 3.0e, 06 December 2023 [CPP_ND_V3.0E]
  - PP-Configuration for Network Devices and MACsec Ethernet Encryption, 2023-03-29 [MOD_MACsec V1.0]
- Additional Package is claimed as follows:
  - Functional Package for SSH Version 1.0, May 13, 2021[PKG_SSH_v1.0].

2.3 CONFORMANCE RATIONALE

This ST provides exact conformance to the items listed in the previous section. The security problem definition, security objectives, and security requirements in this ST are all taken from CFG_NDcPP-MACsec_v2.0 and PKG_SSH_v1.0 performing only the operations defined there.

2.3.1 TECHNICAL DECISIONS

All NIAP TDs issued to date and applicable to NDcPP_v3.0e, PKG_SSH_v1.0 and MOD_MACsec_v1.0 have been considered. Table 5 identifies all applicable TDs.

Table 5 – Relevant Technical Decisions

Technical Decision	Applicable (Y/N)	Applicable cPP/MOD/PKG	Exclusion Rationale (if applicable)
TD0923 – NIT Technical Decision: Auditable event for FAU_STG_EXT.1 and FAU_GEN.1.2	Y	NDcPP_v3.0e	TD is applicable but not relevant since TOE does allow the administrator to configure local audit settings.
TD0921 – NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction Assignment	Y	NDcPP_v3.0e	N/A
TD0900 – NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	Y	NDcPP_v3.0e	N/A
TD0899 – NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2	N	NDcPP_v3.0e	TLSC not part of evaluation.
TD0886 - Clarification to FAU_STG_EXT.1 Test 6	Y	NDcPP_v3.0e	N/A
TD0880 - NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1	Y	NDcPP_v3.0e	N/A
TD0879 - NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E	Y	NDcPP_v3.0e	N/A
TD0868 - NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	N	NDcPP_v3.0e	IPSec is not claimed by the TOE for this evaluation.
TD0836 - NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1	Y	NDcPP_v3.0e	N/A
TD0939 - Updated Conformance Claims for MOD_MACSEC	Y	MOD_MACsec_v1.0	N/A
TD0891 - Correlation of Implicitly Satisfied Requirements when CPP_ND_V3.0E is the Base-PP	Y	MOD_MACsec_v1.0	N/A
TD0889 - Correction For Tests Incorrectly Requiring Group MACsec	Y	MOD_MACsec_v1.0	N/A
TD0884 - Expansion of Permitted EtherTypes in FCS_MACSEC_EXT.1.4	Y	MOD_MACsec_v1.0	N/A
TD0882 - MACsec Data Delay Protection, Key Agreement, and Conditional Support for Group CAK	Y	MOD_MACsec_v1.0	N/A

Technical Decision	Applicable (Y/N)	Applicable cPP/MOD/PKG	Exclusion Rationale (if applicable)
TD0881 - Correction to MN Usage for FPT_RPL.1 Test	Y	MOD_MACsec_v1.0	N/A
TD0870 - Security Objectives Rationale for MOD_MACSEC_V1.0	Y	MOD_MACsec_v1.0	N/A
TD0840 - Alignment of Test 22.1 to FMT_SMF.1/MACSEC	Y	MOD_MACsec_v1.0	N/A
TD0826 - Aligning MOD_MACSEC_V1.0 with CPP_ND_V3.0E	Y	MOD_MACsec_v1.0	N/A
TD0825 - Correction to IEEE 802.1X Reference	Y	MOD_MACsec_v1.0	N/A
TD0816 - Clarity for MACsec Self-Test Failure Response	Y	MOD_MACsec_v1.0	N/A
TD0803 - Clarification for Configurable MACsec CKN Length	Y	MOD_MACsec_v1.0	N/A
TD0746 - Correction to FPT_RPL.1 Test 25	Y	MOD_MACsec_v1.0	N/A
TD0728 - Corrections to MACSec PP-Module SD	Y	MOD_MACsec_v1.0	N/A
TD0967 - Allowance of Kex-strict in PKG_SSH_V1.0	Y	PKG_SSH_v1.0	N/A
TD0909 – Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	Y	PKG_SSH_v1.0	N/A
TD0777 - Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Y	PKG_SSH_v1.0	N/A
TD0732 - FCS_SSHS_EXT.1.3 Test 2 Update	Y	PKG_SSH_v1.0	N/A
TD0695 - Choice of 128- or 256-bit size in AES-CTR in SSH Functional Package.	Y	PKG_SSH_v1.0	N/A
TD0682 - Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	Y	PKG_SSH_v1.0	N/A

3. SECURITY PROBLEM DEFINITION

The security problem definition has been taken directly from the claimed PP, Module and Package specified in Section 2.2 and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any Organizational Security Policies (OSPs) that the TOE is expected to enforce.

3.1 THREATS

The threats included in Table 6 are drawn directly from the PP, Module and Package specified in Section 2.2.

Table 6 – Threats

ID	Threat
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could

ID	Threat
	insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker. Threat agents may also be able to take advantage of weak administrative passwords to gain privileged access to the device.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.
T.DATA_INTEGRITY	An attacker may modify data transmitted over the layer 2 link in a way that is not detected by the recipient. Devices on a network may be exposed to attacks that attempt to corrupt or modify data in transit without authorization. If malicious devices are able to modify and replay data that is transmitted over a layer 2 link, then the data contained within the communications may be susceptible to a loss of integrity.
T.NETWORK_ACCESS	An attacker may send traffic through the TOE that enables them to access devices in the TOE's operational environment without authorization. A MACsec device may

ID	Threat
	sit on the periphery of a network, which means that it may have an externallyfacing interface to a public network. Devices located in the public network may attempt to exercise services located on the internal network that are intended to be accessed only from within the internal network or externally accessible only from specifically authorized devices. If the MACsec device allows unauthorized external devices access to the internal network, these devices on the internal network may be subject to compromise. Similarly, if two MACsec devices are deployed to facilitate end-to-end encryption of traffic that is contained within a single network, an attacker could use an insecure MACsec device as a method to access devices on a specific segment of that network such as an individual LAN.
T.UNTRUSTED_MACSEC_COMMUNICATION_CHANNELS	An attacker may acquire sensitive TOE or user data that is transmitted to or from the TOE because an untrusted communication channel causes a disclosure of data in transit. A generic network device may be threatened by the use of insecure communications channels to transmit sensitive data. The attack surface of a MACsec device also includes the MACsec trusted channels. Inability to secure communications channels, or failure to do so correctly, would expose user data that is assumed to be secure to the threat of unauthorized disclosure.

3.2 ASSUMPTIONS

The assumptions included in Table 7 are drawn directly from PP, Module and Package.

Table 7 – Assumptions

ID	Assumption
A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.

ID	Assumption
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).
A.NO_THRU_TRAFFIC_PROTECTION	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.
A.REGULAR_UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The Administrator’s credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

3.3 ORGANIZATIONAL SECURITY POLICIES

The OSPs included in Table 8 are drawn directly from the PP, Module and Package.

Table 8 – OSPs

ID	OSP
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which Administrators consent by accessing the TOE.

4. SECURITY OBJECTIVES

The security objectives have been taken directly from the claimed PP, Module and Package and are reproduced here for the convenience of the reader.

4.1 SECURITY OBJECTIVES FOR THE TOE

The security objectives in the following table apply to the TOE. These are augmented by the statement of security objectives for the TOE in relation to the MACsec capabilities as detailed in MOD_MACsec_v1.0.

Table 9 – Security Objectives

ID	Security Objectives
O.AUTHENTICATION_MACSEC	To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (MKA) will allow a MACsec peer to establish connectivity associations (CAs) with another MACsec peer. MACsec endpoints authenticate each other to ensure they are communicating with an authorized MAC Security Entity (SecY) entity
O.AUTHORIZED_ADMINISTRATION	All network devices are expected to provide services that allow the security functionality of the device to be managed. The MACsec device, as a specific type of network device, has a refined set of management functions to address its specialized behavior. In order to further mitigate the threat of a compromise of its security functionality, the MACsec device prescribes the ability to limit brute-force authentication attempts by enforcing lockout of accounts that experience excessive failures and by limiting access to security-relevant data that administrators do not need to view
O.CRYPTOGRAPHIC_FUNCTIONS_MACSEC	To address the issues associated with unauthorized modification and disclosure of information, compliant TOEs will implement cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.
O.PORT_FILTERING_MACSEC	To further address the issues associated with unauthorized network access, a compliant TOE's port filtering capability will restrict the flow of network traffic through the TOE based on layer 2 frame characteristics and whether or not the traffic represents valid MACsec frames and MACsec Key Agreement Protocol Data Units (MKPDUs).
O.REPLAY_DETECTION	A MACsec device is expected to help mitigate the threat of MACsec data integrity violations by providing a

ID	Security Objectives
	mechanism to detect and discard replayed traffic for MPDUs.
O.SYSTEM_MONITORING_MACSEC	To address the issues of administrators being able to monitor the operations of the MACsec device, compliant TOEs will implement the ability to log the flow of Ethernet traffic. Specifically, the TOE will provide the means for administrators to configure rules to 'log' when Ethernet traffic grants or restricts access. As a result, the 'log' will result in informative event logs whenever a match occurs. In addition, the establishment of security CAs is auditable, not only between MACsec devices, but also with MAC Security Key Agreement Entities (KaYs).
O.TSF_INTEGRITY	To mitigate the security risk that the MACsec device may fail during startup, it is required to fail-secure if any self-test failures occur during startup. This ensures that the device will only operate when it is in a known state.

4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

Security objectives for the operational environment assist the TOE in correctly providing its security functionality. These objectives, which are found in the table below, track the assumptions about the TOE operational environment.

Table 10 – Security Objectives for the Operational Environment

ID	Objectives for the Operational Environment
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

ID	Objectives for the Operational Environment
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

5. SECURITY REQUIREMENTS

This section identifies the Security Functional Requirements (SFRs) for the TOE. The SFRs included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revisions 5, April 2017, and all international interpretations.

Table 11 – SFRs

Requirement	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.1/MACSEC	Audit Data Generation (MACsec)
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_COP.1/CMAC	Cryptographic Operation (AES-CMAC Keyed Hash Algorithm)
FCS_COP.1/MACSEC	Cryptographic Operation (MACsec AES Data Encryption and Decryption)
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSH_EXT.1	SSH Protocol
FCS_SSHS_EXT.1	SSH Protocol - Server
FCS_MACSEC_EXT.1	MACsec
FCS_MACSEC_EXT.2	MACsec Integrity and Confidentiality
FCS_MACSEC_EXT.3	MACsec Randomness
FCS_MACSEC_EXT.4	MACsec Key Usage
FCS_MKA_EXT.1	MACsec Key Agreement
FIA_AFL.1	Authentication Failure Handling (Refinement)
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU.7	Protected Authentication Feedback (Refinement)
FIA_PSK_EXT.1	Pre-Shared Key Composition
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour
FMT_MOF.1/Functions	Management of Security Functions Behaviour
FMT_MOF.1/Services	Management of Security Functions Behaviour
FMT_MTD.1/CoreData	Management of TSF Data

Requirement	Description
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMF.1/MACSEC	Specification of Management Functions (MACsec)
FMT_SMR.2	Restrictions on Security Roles
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF Testing (Extended)
FPT_TUD_EXT.1	Trusted Update
FPT_STM_EXT.1	Reliable Time Stamps
FPT_CAK_EXT.1	Protection of CAK Data
FPT_FLS.1	Failure with Preservation of Secure State
FPT_RPL.1	Replay Detection
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_TAB.1	Default TOE Access Banners (Refinement)
FTP_ITC.1	Inter-TSF Trusted Channel (Refinement)
FTP_TRP.1/Admin	Trusted Path (Refinement)
FTP_ITC.1/MACSEC	Inter-TSF Trusted Channel (MACsec Communications)

5.1 CONVENTIONS

The CC allows the following types of operations to be performed on the functional requirements: assignments, selections, refinements, and iterations. The following font conventions are used within this document to identify operations defined by CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration identifier after a slash, e.g., /SigGen.
- Where operations were completed in the cPP and relevant EPs/Modules/Packages/TDs, the formatting used in the source document has been retained except for text within brackets. If the text within brackets is completing an operation by the ST author, the conventions from the first three bullets apply. Otherwise, the text is presented in plaintext.
- Extended SFRs are identified by the addition of “EXT” after the requirement name.

5.2 SECURITY FUNCTIONAL REQUIREMENTS

This section includes the security functional requirements for this ST.

5.2.1 SECURITY AUDIT (FAU)

5.2.1.1 FAU_GEN.1 AUDIT DATA GENERATION

FAU_GEN.1.1

The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;

b) All auditable events for the not specified level of audit; and

c) *All administrative actions comprising:*

- *Administrative login and logout (name of Administrator account shall be logged if individual accounts are required for Administrators).*
- *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
- [Resetting passwords (name of related Administrator account shall be logged)]*;*

d) *Specifically defined auditable events listed in* **Table 12***.*

FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity ~~(if applicable),~~ and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of* **Table 12***.*

Table 12 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
FCS_CKM.1	None	None
FCS_CKM.2	None	None
FCS_CKM.4	None	None
FCS_COP.1/DataEncryption	None	None
FCS_COP.1/SigGen	None	None
FCS_COP.1/Hash	None	None
FCS_COP.1/KeyedHash	None	None
FCS_RBG_EXT.1	None	None
FCS_SSH_EXT.1 **[TD0777]**	Failure to establish SSH connection	Reason for failure and Non-TOE endpoint of attempted connection (IP Address)
	Establishment of SSH connection	None

Requirement	Auditable Events	Additional Audit Record Contents
	Termination of SSH connection session	None
	Dropping of packet(s) outside defined size limits	None
FIA_UIA_EXT.1	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded	Origin of the attempt (e.g., IP address)
FIA_PMG_EXT.1	None	None
FIA_UAU.7	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None
FMT_MOF.1/Functions	None	None
FMT_MOF.1/Services	None	None
FMT_MTD.1/CoreData	None	None
FMT_MTD.1/CryptoKeys	None	None
FMT_SMF.1	All management activities of TSF data.	None
FMT_SMR.2	None	None
FPT_SKP_EXT.1	None	None
FPT_APW_EXT.1	None	None
FPT_TST_EXT.1	None	None
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL.3	The termination of a remote session by the session locking mechanism	None
FTA_SSL.4	The termination of an interactive session	None
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session locking mechanism	None
FTA_TAB.1	None	None

Requirement	Auditable Events	Additional Audit Record Contents
FTP_ITC.1	• Initiation of the trusted channel • Termination of the trusted channel • Failure of the trusted channel functions	• None • None • Reason for failure
FTP_TRP.1/Admin	• Initiation of the trusted path • Termination of the trusted path. • Failure of the trusted path functions.	• None • None • Reason for failure

Application Note: TD0923 is applicable but not relevant since the TOE does allow the administrator to configure local audit settings.

5.2.1.2 FAU_GEN.1/MACSEC AUDIT DATA GENERATION (MACSEC)

FAU_GEN.1.1/MACSEC

The TSF shall be able to generate an audit record of the following auditable events:

a. Start-up and shutdown of the audit functions;
b. All auditable events for the [*not specified*] level of audit;
c. **All administrative actions;**
d. [Specifically defined auditable events listed in the Auditable Events table (Table 13)]

Table 13 – Security Functional Requirements and Auditable Events/MACsec

Requirement	Auditable Events	Additional Audit Record Contents
FCS_MACSEC_EXT.1	Session establishment	Secure Channel Identifier (SCI)
FCS_MACSEC_EXT.3	Creation and update of SAK	Creation and update times
FCS_MACSEC_EXT.4	Creation of CA	Connectivity Association Key Names (CKNs)
FPT_RPL.1	Detected replay attempt	None

FAU_GEN.1.2/MACSEC

The TSF shall record within each audit record at least the following information:

a. Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
b. For each audit event type, based on the auditable event definitions of the functional components included in the PP-**Module**/ST, [information specified in column three of the Auditable Events table (Table 13)].

5.2.1.3 FAU_GEN.2 USER IDENTITY ASSOCIATION

FAU_GEN.2.1
For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.1.4 FAU_STG_EXT.1 PROTECTED AUDIT EVENT STORAGE

FAU_STG_EXT.1.1
The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2
The TSF shall be able to store generated audit data on the TOE itself. In addition [

- The TOE shall consist of a single standalone component that stores audit data locally

].

FAU_STG_EXT.1.3
The TSF shall maintain a [log file] of audit records in the event that an interruption of communication with the remote audit server occurs.

FAU_STG_EXT.1.4
The TSF shall be able to store [persistent] audit records locally with a minimum storage size of [*64KB file size*].

FAU_STG_EXT.1.5
The TSF shall [overwrite previous audit records according to the following rule: *[oldest log file is overwritten]*] when the local storage space for audit data is full.

FAU_STG_EXT.1.6

The TSF shall provide the following mechanisms for administrative access to locally stored audit records [manual export, ability to view locally].

5.2.2 CRYPTOGRAPHIC SUPPORT (FCS)

5.2.2.1 FCS_CKM.1 CRYPTOGRAPHIC KEY GENERATION

FCS_CKM.1.1 [TD0921]
The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- RSA schemes using cryptographic key sizes of [*2048, 3072, 4096 bits*] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;
- ECC schemes using ‘NIST curves’ [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4 or FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Appendix A.2, or ISO/IEC 14888-3, “IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms”, Section 6.6.;

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: *list of standards*].~~

5.2.2.2 FCS_CKM.2 CRYPTOGRAPHIC KEY ESTABLISHMENT

FCS_CKM.2.1

The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";

] ~~that meets the following: [assignment: *list of standards*].~~

5.2.2.3 FCS_CKM.4 CRYPTOGRAPHIC KEY DESTRUCTION

FCS_CKM.4.1

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- *For plaintext keys in volatile storage, the destruction shall be executed by a* [destruction of reference to the key directly followed by a request for garbage collection];
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that* [
  - logically addresses the storage location of the key and performs a [single overwrite consisting of [zeroes]];

that meets the following: *No Standard.*

5.2.2.4 FCS_COP.1/CMAC CRYPTOGRAPHIC OPERATION (AES-CMAC KEYED HASH ALGORITHM)

FCS_COP.1.1/CMAC

The TSF shall perform [keyed-hash message authentication] in accordance with a specified cryptographic algorithm [*AES-CMAC*] and cryptographic key sizes [128, 256] **bits and message digest size of 128 bits** that meets the following: [NIST SP 800-38B].

5.2.2.5 FCS_COP.1/DATAENCRYPTION CRYPTOGRAPHIC OPERATIONS (AES DATA ENCRYPTION/DECRYPTION)

FCS_COP.1.1/DataEncryption

The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* [CBC, CTR, GCM] *mode* and cryptographic key sizes [128 bits, 256 bits] that meet the following: *AES as specified in ISO 18033-3,* [CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772].

5.2.2.6 FCS_COP.1/HASH CRYPTOGRAPHIC OPERATIONS (HASH ALGORITHM)

FCS_COP.1.1/Hash
The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512] ~~and cryptographic key sizes~~ [~~*assignment: cryptographic key sizes*~~] and **message digest sizes** [160, 256, 384, 512] **bits** that meet the following: *ISO/IEC 10118-3:2004*.

5.2.2.7 FCS_COP.1/KEYEDHASH CRYPTOGRAPHIC OPERATION (KEYED HASH ALGORITHM)

FCS_COP.1.1/KeyedHash
The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [HMAC-SHA-256, HMAC-SHA-512] and cryptographic key sizes *[256 and 512 bits]* **and message digest sizes** [256, 512] ***bits*** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.2.2.8 FCS_COP.1/MACSEC CRYPTOGRAPHIC OPERATION (MACSEC AES DATA ENCRYPTION AND DECRYPTION)

FCS_COP.1.1/MACSEC
The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES used in AES Key Wrap, GCM] and cryptographic key sizes [128, 256] **bits** that meets the following: [AES as specified in ISO 18033-3, AES Key Wrap as specified in NIST SP 800- 38F, GCM as specified in ISO 19772].

5.2.2.9 FCS_COP.1/SIGGEN CRYPTOGRAPHIC OPERATION (SIGNATURE GENERATION AND VERIFICATION)

FCS_COP.1.1/SigGen [TD0921]
The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm,
- Elliptic Curve Digital Signature Algorithm

]
and cryptographic key sizes [

- For RSA: modulus 2048, 3072, 4096 bits,
- For ECDSA: 256 bits, 384 and 521 bits

]
that meet the following: [

- For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,
- For ECDSA schemes: implementing [P-256, P-384, P-521] curves that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST Recommended" curves; or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.

].

5.2.2.10 FCS_MACSEC_EXT.1 MACSEC

FCS_MACSEC_EXT.1.1
The TSF shall implement MACsec in accordance with IEEE Standard 802.1AE-2018.

FCS_MACSEC_EXT.1.2
The TSF shall derive a Secure Channel Identifier (SCI) from a peer's MAC address and port to uniquely identify the originator of an MPDU.

FCS_MACSEC_EXT.1.3
The TSF shall reject any MPDUs during a given session that contain an SCI other than the one used to establish that session.

FCS_MACSEC_EXT.1.4 [TD0884]
The TSF shall permit only EAPOL (Port Access Entity (PAE) EtherType 88-8E), MACsec frames (EtherType 88-E5), and [MAC control frames (EtherType is 88- 08)] and shall discard others.

5.2.2.11 FCS_MACSEC_EXT.2 MACSEC INTEGRITY AND CONFIDENTIALITY

FCS_MACSEC_EXT.2.1
The TOE shall implement MACsec with support for integrity protection with a confidentiality offset of [0, 30, 50].

FCS_MACSEC_EXT.2.2
The TSF shall provide assurance of the integrity of protocol data units (MPDUs) using an Integrity Check Value (ICV) derived with the SAK.

FCS_MACSEC_EXT.2.3
The TSF shall provide the ability to derive an Integrity Check Value Key (ICK) from a Connectivity Association Key (CAK) using a KDF.

5.2.2.12 FCS_MACSEC_EXT.3 MACSEC RANDOMNESS

FCS_MACSEC_EXT.3.1 [TD0825]
The TSF shall generate unique Secure Association Keys (SAKs) using [key derivation from Connectivity Association Key (CAK) per section 9.8.1 of IEEE 802.1X-2020] such that the likelihood of a repeating SAK is no less than 1 in 2 to the power of the size of the generated key.

FCS_MACSEC_EXT.3.2
The TSF shall generate unique nonces for the derivation of SAKs using the TOE's random bit generator as specified by FCS_RBG_EXT.1.

5.2.2.13 FCS_MACSEC_EXT.4 MACSEC KEY USAGE

FCS_MACSEC_EXT.4.1
The TSF shall support peer authentication using pre-shared keys (PSKs) [no other method].

FCS_MACSEC_EXT.4.2
The TSF shall distribute SAKs between MACsec peers using AES key wrap as specified in FCS_COP.1/**MACSEC**.

FCS_MACSEC_EXT.4.3
The TSF shall support specifying a lifetime for CAKs.

FCS_MACSEC_EXT.4.4 [TD0825]
The TSF shall associate Connectivity Association Key Names (CKNs) with SAKs that are defined by the KDF using the CAK as input data (per IEEE 802.1X-2020, Section 9.8.1).

FCS_MACSEC_EXT.4.5
The TSF shall associate CKNs with CAKs. The length of the CKN shall be an integer number of octets, between 1 and 32 (inclusive).

5.2.2.14 FCS_MKA_EXT.1 MACSEC KEY AGREEMENT

FCS_MKA_EXT.1.1 [TD0825]
The TSF shall implement Key Agreement Protocol (MKA) in accordance with IEEE 802.1X-2020 and 802.1Xbx-2014.

FCS_MKA_EXT.1.2
The TSF shall provide assurance of the integrity of MKA protocol data units (MKPDUs) using an Integrity Check Value (ICV) derived from an Integrity Check Value Key (ICK).

FCS_MKA_EXT.1.3
The TSF shall provide the ability to derive an Integrity Check Value Key (ICK) from a CAK using a KDF.

FCS_MKA_EXT.1.4 [TD0882]
The TSF shall enforce an MKA Lifetime Timeout limit of 6.0 seconds and [MKA Hello Time limit of 2 seconds].

FCS_MKA_EXT.1.5
The key server shall refresh a SAK when it expires. The key server shall distribute a SAK by [pairwise CAKs that are PSKs].

FCS_MKA_EXT.1.6
The key server shall distribute a fresh SAK whenever a member is added to or removed from the live membership of the CA.

FCS_MKA_EXT.1.7 [TD0825]
The TSF shall validate MKPDUs according to IEEE 802.1X-2020 Section 11.11.2. In particular, the TSF shall discard without further processing any MKPDUs to which any of the following conditions apply:

a. The destination address of the MKPDU was an individual address.
b. The MKPDU is less than 32 octets long.
c. The MKPDU comprises fewer octets than indicated by the Basic Parameter Set body length, as encoded in bits 4 through 1 of octet 3 and bits 8 through 1 of octet 4, plus 16 octets of ICV.
d. The CAK Name is not recognized.

If an MKPDU passes these tests, then the TSF will begin processing it as follows:

a. If the Algorithm Agility parameter identifies an algorithm that has been implemented by the receiver, the ICV shall be verified as specified in IEEE 802.1X-2020 Section 9.4.1.
b. If the Algorithm Agility parameter is unrecognized or not implemented by the receiver, its value can be recorded for diagnosis but the received MKPDU shall be discarded without further processing.

Each received MKPDU that is validated as specified in this clause and verified as specified in IEEE 802.1X-2020 Section 9.4.1 shall be decoded as specified in IEEE 802.1X-2020 Section 11.11.4.

5.2.2.15 FCS_RBG_EXT.1 RANDOM BIT GENERATION

FCS_RBG_EXT.1.1
The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [HMAC_DRBG [SHA-512]].

FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[1]* software-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

5.2.2.16 FCS_SSH_EXT.1 SSH PROTOCOL

FCS_SSH_EXT.1.1 [TD0909]
The TOE shall implement SSH acting as a [server] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [4256,4344, 5656, 6668, 8308, 8332] and *[no other standard].*

FCS_SSH_EXT.1.2
The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [

- "password" (RFC 4252),
- "keyboard-interactive" (RFC 4256),
- "publickey" (RFC 4252): [
  - ssh-rsa (RFC 4253),
  - rsa-sha2-256 (RFC 8332),
  - rsa-sha2-512 (RFC 8332),
  - ecdsa-sha2-nistp256 (RFC 5656),
  - ecdsa-sha2-nistp384 (RFC 5656),
  - ecdsa-sha2-nistp521 (RFC 5656),

  ]

] and no other methods.

FCS_SSH_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than [*256K bytes*] in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4

The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [

- aes128-ctr (RFC 4344),
- aes256-ctr (RFC 4344),
- aes128-cbc (RFC 4253),
- aes256-cbc (RFC 4253)]

and no other mechanisms.

FCS_SSH_EXT.1.5

The TSF shall protect data in transit from modification, deletion, and insertion using: [

- hmac-sha2-256 (RFC 6668),
- hmac-sha2-512 (RFC 6668)]

and no other mechanisms.

FCS_SSH_EXT.1.6

The TSF shall establish a shared secret with its peer using: [

- ecdh-sha2-nistp256 (RFC 5656),
- ecdh-sha2-nistp384 (RFC 5656),
- ecdh-sha2-nistp521 (RFC 5656)]

and no other mechanisms.

FCS_SSH_EXT.1.7

The TSF shall use SSH KDF as defined in [RFC 5656 (Section 4)] to derive the following cryptographic keys from a shared secret: session keys.

FCS_SSH_EXT.1.8

The TSF shall ensure that [a rekey of the session keys] occurs when any of the following thresholds are met: one hour connection time no more than one gigabyte of transmitted data, or no more than one gigabyte of received data.

5.2.2.17 FCS_SSHS_EXT.1 SSH PROTOCOL – SERVER

FCS_SSHS_EXT.1.1

The TSF shall authenticate itself to its peer (SSH Client) using: [

- ssh-rsa (RFC 4253),
- rsa-sha2-256 (RFC 8332),
- rsa-sha2-512 (RFC 8332),
- ecdsa-sha2-nistp256 (RFC 5656),
- ecdsa-sha2-nistp384 (RFC 5656),
- ecdsa-sha2-nistp521 (RFC 5656

].

5.2.3 IDENTIFICATION AND AUTHENTICATION (FIA)

5.2.3.1 FIA_AFL.1 AUTHENTICATION FAILURE HANDLING

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within [1 to 10] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until [*an authorized Administrator unlocks the locked user account*] is taken by an Administrator; prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].

5.2.3.2 FIA_PMG_EXT.1 PASSWORD MANAGEMENT

FIA_PMG_EXT.1.1
The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers and the following special characters: [*"!", "@", "#", "$", "%", "^", "&", "*", "(", ")",["~", "`", ",", ".", "/", ":", ";", " ", "+", "-", "=", "{", "}", "[", "]", "|", "<", ">"]*];

b) Minimum password length shall be *configurable to between [6] and [20] characters.*

5.2.3.3 FIA_PSK_EXT.1 PRE-SHARED KEY COMPOSITION

FIA_PSK_EXT.1.1 [TD0825]
The TSF shall use PSKs for MKA as defined by IEEE 802.1X-2020, [no other protocols].

FIA_PSK_EXT.1.2
The TSF shall be able to [accept] bit-based PSKs.

5.2.3.4 FIA_UIA_EXT.1 USER IDENTIFICATION AND AUTHENTICATION

FIA_UIA_EXT.1.1
The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [*[ICMP echo replies]*].

FIA_UIA_EXT.1.2
The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

FIA_UIA_EXT.1.3 [TD0900]
The TSF shall provide the following remote authentication mechanisms [SSH password, SSH public key] and [*no other mechanism*]. The TSF shall provide the following local authentication mechanisms [password-based].

FIA_UIA_EXT.1.4
The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

5.2.3.5 FIA_UAU.7 PROTECTED AUTHENTICATION FEEDBACK (REFINEMENT)

FIA_UAU.7.1
The TSF shall provide only *obscured feedback* to the **administrative** user while the authentication is in progress **at the local console**.

5.2.4 SECURITY MANAGEMENT (FMT)

5.2.4.1 FMT_MOF.1/MANUALUPDATE MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOR

FMT_MOF.1.1/ManualUpdate
The TSF shall restrict the ability to enable the functions *to perform manual updates* to *Security Administrators.*

5.2.4.2 FMT_MOF.1/FUNCTIONS MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR

FMT_MOF.1.1/Functions
The TSF shall restrict the ability to [modify the behaviour of] the functions [transmission of audit data to an external IT entity, handling of audit data] to *Security Administrators*.

5.2.4.3 FMT_MOF.1/SERVICES MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR

FMT_MOF.1.1/Services
The TSF shall restrict the ability to **start and stop** ~~the functions~~ **services** to *Security Administrators*.

5.2.4.4 FMT_MTD.1/COREDATA MANAGEMENT OF TSF DATA

FMT_MTD.1.1/CoreData
The TSF shall restrict the ability to *manage* the *TSF data* to *Security Administrators*.

5.2.4.5 FMT_MTD.1/CRYPTOKEYS MANAGEMENT OF TSF DATA

FMT_MTD.1.1/CryptoKeys
The TSF shall restrict the ability to *manage* the *cryptographic keys* to *Security Administrators*.

5.2.4.6 FMT_SMF.1 SPECIFICATION OF MANAGEMENT FUNCTIONS

FMT_SMF.1.1 [TD0880]
The TSF shall be capable of performing the following management functions:

- *Ability to administer the TOE remotely;*

- *Ability to configure the access banner;*
- *Ability to configure the remote session inactivity time before session termination;*
- *Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;*
- [
    - Ability to start and stop services;
    - Ability to modify the behaviour of the transmission of audit data to an external IT entity
    - Ability to configure local audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full, changes to local audit storage size);
    - Ability to manage the cryptographic keys;
    - Ability to manage the cryptographic functionality;
    - Ability to configure thresholds for SSH rekeying;
    - Ability to re-enable an Administrator account;
    - Ability to set the time which is used for time-stamps;
    - Ability to administer the TOE locally;
    - Ability to configure the local session inactivity time before session termination or locking;
    - Ability to configure the authentication failure parameters for FIA_AFL.1;
    - Ability to manage the trusted public keys database;].

5.2.4.7 FMT_SMF.1/MACSEC SPECIFICATION OF MANAGEMENT FUNCTIONS (MACSEC)

FMT_SMF.1.1/MACSEC
The TSF shall be capable of performing the following management functions **related to MACsec functionality**:
[Ability of a Security Administrator to:

- Manage a PSK-based CAK and install it in the device
- Manage the key server to create, delete, and activate MKA participants [*[Command Line Interface commands]*]
- Specify the lifetime of a CAK
- Enable, disable, or delete a PSK-based CAK using [*[Command Line Interface commands]*]

[

- No other MACsec management functions

]

].

5.2.4.8 FMT_SMR.2 RESTRICTIONS ON SECURITY ROLES

FMT_SMR.2.1
The TSF shall maintain the roles:

- *Security Administrator.*

FMT_SMR.2.2
The TSF shall be able to associate users with roles.

FMT_SMR.2.3
The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE remotely* are satisfied.

5.2.5 PROTECTION OF THE TSF (FPT)

5.2.5.1 FPT_APW_EXT.1 PROTECTION OF ADMINISTRATOR PASSWORDS

FPT_APW_EXT.1.1
The TSF shall store administrative passwords in non-plaintext form.

FPT_APW_EXT.1.2
The TSF shall prevent the reading of plaintext administrative passwords.

5.2.5.2 FPT_CAK_EXT.1 PROTECTION OF CAK DATA

FPT_CAK_EXT.1.1

The TSF shall prevent reading of CAK values by administrators.

5.2.5.3 FPT_FLS.1 FAILURE WITH PRESERVATION OF SECURE STATE

FPT_FLS.1.1
The TSF shall **fail-secure** when **any of** the following types of failures occur: [failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests].

5.2.5.4 FPT_RPL.1 REPLAY DETECTION

FPT_RPL.1.1
The TSF shall detect replay for the following entities: [MPDUs, MKA frames].

FPT_RPL.1.2
The TSF shall perform [discarding of the replayed data, logging of the detected replay attempt] when replay is detected.

5.2.5.5 FPT_SKP_EXT.1 PROTECTION OF TSF DATA (FOR READING OF ALL SYMMETRIC KEYS)

FPT_SKP_EXT.1.1
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.2.5.6 FPT_STM_EXT.1 RELIABLE TIME STAMPS

FPT_STM_EXT.1.1
The TSF shall be able to provide reliable time stamps for its own use.

FPT_STM_EXT.1.2
The TSF shall [allow the Security Administrator to set the time].

5.2.5.7 FPT_TST_EXT.1 TSF TESTING

FPT_TST_EXT.1.1 [TD0836]
The TSF shall run a suite of the following self-tests:

- During initial start-up (on power on) to verify the integrity of the TOE firmware and software;
- Prior to providing any cryptographic service and [on-demand] to verify correct operation of cryptographic implementation necessary to fulfil the TSF;
- [at the conditions [*Noise Source Health Test*]] self-tests.

to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2
The TSF shall respond to [all failures] by [*Rebooting*].

5.2.5.8 FPT_TUD_EXT.1 TRUSTED UPDATE

FPT_TUD_EXT.1.1
The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [no other TOE firmware/software version].

FPT_TUD_EXT.1.2
The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [no other update mechanism].

FPT_TUD_EXT.1.3
The TSF shall provide means to authenticate firmware/software updates to the TOE using a [digital signature] prior to installing those updates.

5.2.6 TOE ACCESS (FTA)

5.2.6.1 FTA_SSL.3 TSF-INITIATED TERMINATION

FTA_SSL.3.1
The TSF shall terminate **a remote** interactive session after a *Security Administrator-configurable time interval of session inactivity.*

5.2.6.2 FTA_SSL.4 USER-INITIATED TERMINATION

FTA_SSL.4.1
The TSF shall allow ~~user~~ **Administrator**-initiated termination of the ~~user's~~ **Administrator's** own interactive session.

5.2.6.3 FTA_SSL_EXT.1 TSF-INITIATED SESSION LOCKING

FTA_SSL_EXT.1.1
The TSF shall, for local interactive sessions, [

- terminate the session]

after a Security Administrator-specified time period of inactivity.

5.2.6.4 FTA_TAB.1 DEFAULT TOE ACCESS BANNERS

FTA_TAB.1.1
Before establishing ~~a~~ **an administrative** user session the TSF shall display **a Security Administrator-specified** advisory **notice and consent** warning message regarding ~~unauthorised~~ use of the TOE.

5.2.7 TRUSTED PATH/CHANNELS (FTP)

5.2.7.1 FTP_ITC.1 INTER-TSF TRUSTED CHANNEL

FTP_ITC.1.1
The TSF shall **be capable of using [SSH] to** provide a **trusted** communication channel between itself and ~~another trusted IT product~~ **authorized IT entities supporting the following capabilities: audit server,** [no other capabilities] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from ~~modification or~~ disclosure **and detection of modification of the channel data**.

FTP_ITC.1.2
The TSF shall permit [the authorized IT entities] to initiate communication via the trusted channel.

FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for [*no services*].

5.2.7.2 FTP_ITC.1/MACSEC INTER-TSF TRUSTED CHANNEL (MACSEC COMMUNICATIONS)

FTP_ITC.1.1/MACSEC
The TSF shall provide a communication channel between itself and **a MACsec peer** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/MACSEC
The TSF shall permit [the TSF, another trusted IT product] to initiate communication via the trusted channel.

FTP_ITC.1.3/MACSEC
The TSF shall initiate communication via the trusted channel for [communications with MACsec peers that require the use of MACsec].

5.2.7.3 FTP_TRP.1/ADMIN TRUSTED PATH

FTP_TRP.1.1/Admin
The TSF shall **be capable of using [SSH] to** provide a communication path between itself and **authorized** remote **Administrators** ~~users~~ that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure **and provides detection of modification of the channel data**.

FTP_TRP.1.2/Admin
The TSF shall permit remote **Administrators** ~~users~~ to initiate communication via the trusted path.

FTP_TRP.1.3/Admin
The TSF shall require the use of the trusted path for *initial Administrator authentication and all remote administration actions*.

5.3 TOE SFR DEPENDENCIES RATIONALE FOR SFRS

The PP and any relevant Module/Package contain(s) all the requirements claimed in this ST. As such, the dependencies are not applicable since the PP has been approved.

5.4 SECURITY ASSURANCE REQUIREMENTS

The TOE assurance requirements for this ST are taken directly from the PP, Module and Package, which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in Table 14.

Table 14 – Security Assurance Requirements

Assurance Class	Assurance Components	Component Description
Security Target	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic functional specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage

Assurance Class	Assurance Components	Component Description
Tests	ATE_IND.1	Independent testing – conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability survey

5.5 ASSURANCE MEASURES

The TOE satisfied the identified assurance requirements. This section identifies the Assurance Measures applied by HPE Juniper Networking to satisfy the assurance requirements. The following table lists the details.

Table 15 – TOE Security Assurance Measures

SAR Component	How the SAR will be met
ASE_TSS.1.1C Refinement	Vendor will provide information on how the TOE meets each claimed SFR. This information has been documented in detail in the subsequent section.
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides descriptions of the processes and procedures on how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ALC_CMS.1	
ATE_IND.1	Vendor will provide the TOE for testing.
AVA_VAN.1	Vendor will provide the TOE for testing. Vendor will provide a document identifying the list of software and hardware components.

6. TOE SUMMARY SPECIFICATION

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

Table 16 – TOE Summary Specification

Requirement	TSS Description
FAU_GEN.1 FAU_GEN.1/MACSEC	The TOE generates and stores a comprehensive set of audit logs that identify specific TOE operation whenever an auditable event occurs. Auditing is implemented using syslog. The Auditable events are specified in Table 12 – Security Functional Requirements and Auditable Events and Table 13 – Security Functional Requirements and Auditable Events/MACsec The TOE records the following with each log entry: • date and time of the event and/or reaction • type of event and/or reaction • subject identity (where applicable) • the outcome (success or failure) of the event (where applicable). In order to identify the key being operated on, the following details are recorded for all administrative actions relating to cryptographic keys (generating, importing, changing and deleting keys): • CAK – imported key reference is recorded in syslog • SAK – Key Identifier is recorded in syslog • KEK, SAK, ICV – key references provided by process id • SSH session keys– key reference provided by process id • SSH key imported for SSH public key authentication –the hash of the public key that is to be used for authentication is recorded in syslog • Username and key type - The key type (rsa/ecdsa) and the username of the associated user will be recorded when importing an SSH user public key
FAU_GEN.2	The TOE ensures that each auditable event is associated with the identity of the user that triggered the event.
FAU_STG_EXT.1	The TOE stores audit logs locally in an audit file. Additionally, syslog can be configured to transmit and store audit data securely via Netconf over SSH. Audit data is transmitted to the configured syslog server in real time. The logs are of persistent nature and these local audit logs are stored in /var/log/ in the underlying filesystem. These audit logs are automatically overwritten as per the administrative configurable limits on storage volume. The default maximum size is 1Gb. Only authorized Security Administrators are allowed to read and delete log files. These log files can be accessed using command-line interface (CLI) or via direct filesystem access, but only after successfully authentication as a Security Administrator. The TOE uses an active log file and archive files which are configurable from 1-1000, default being 10. When the active log file reaches its maximum size, it's compressed and archived (logfile.0.gz), and a new active log file is created. Subsequent full active log files shift the archive file numbering (logfile.0.gz

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td></td><td>becomes logfile.1.gz, etc.). Upon reaching the maximum archive file limit, the oldest archive log file is overwritten to accommodate the newly archived log file.

Note: The TOE is standalone and NOT considered distributed.</td></tr>
<tr><td>FCS_CKM.1</td><td>The TOE uses several cryptographic key generation schemes with different key sizes for each specific operation. The table below details the same:
<table>
<tr><th>Key Generation Scheme</th><th>SFR</th><th>Key Size(s)</th><th>Algorithm</th><th>Usage</th></tr>
<tr><td>SSH Key Generation (RSA)
- FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1</td><td>FCS_CKM.1,
FCS_SSH_EXT.1
FCS_SSHS_EXT.1</td><td>2048-bit, 3072-bit, 4096-bit</td><td>RSA</td><td rowspan="2">Asymmetric key generation with respective hashing algorithms and curves.</td></tr>
<tr><td>SSH Key Generation (ECDSA)
- FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Appendix A.2</td><td>FCS_CKM.1,
FCS_SSHS_EXT.1</td><td>P-256, P-384, P-521</td><td>ECDSA</td></tr>
</table></td></tr>
<tr><td>FCS_CKM.2</td><td>The TOE supports several key establishment and key generation schemes. The table below details the same:
<table>
<tr><th>Scheme</th><th>SFR(s)</th><th>Usage/Service</th></tr>
<tr><td>MACsec Key Agreement (MKA)</td><td>FCS_MKA_EXT.1
FCS_CKM.2</td><td>Secure establishment of MACsec SAKs</td></tr>
<tr><td>SSH EC Key Agreement
- NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”</td><td>FCS_CKM.2</td><td>Secure establishment of SSH session</td></tr>
</table></td></tr>
<tr><td>FCS_CKM.4</td><td>The TOE stores plaintext keys in volatile and non-volatile storage. The TOE satisfies all requirements for destruction of CSPs, plaintext secret and private cryptographic keys as specified in Table 18 – Zeroization of Keys and CSP.

There are no configurations or circumstances that may not conform to the key destruction requirements.</td></tr>
<tr><td>FCS_COP.1/DataEncryption</td><td>The TOE supports AES encryption and decryption conforming to CBC, CTR and GCM as specified in ISO 10116-3, ISO 10116 and ISO 19772 respectively. The AES key sizes supported are 128 and 256 bits. AES is implemented in the following protocols: SSH, MACsec.

Please refer to Table 17 – CAVP Algorithm Certificate References for NIST CAVP certificate numbers for AES.</td></tr>
</table>

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td></td><td></td></tr>
<tr><td>FCS_COP.1/SigGen</td><td>The TOE provides cryptographic signature generation and verification services in accordance with the following cryptographic algorithms that meet the following schemes:
RSA (RSA digital signature algorithm conforming to FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5): Key sizes of 2048, 3072, or 4096 bits
ECDSA (Elliptical curve digital signature algorithm conforming to FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves): Key sizes of 256, 384, or 521 bits</td></tr>
<tr><td>FCS_COP.1/Hash</td><td>The TOE supports cryptographic hashing services in accordance with SHA-1, SHA2-256, SHA2- 384 and SHA2-512 for SSHv2 protocol with message digest size of 160, 256, 384, 512.

Additionally, the TOE also performs hashing while performing image integrity check during update, running self-tests and storing password. Trusted Update signature verification uses ECDSA on P-256 w/SHA-256. The TOE supports following keyed-hash message authentication algorithms HMAC-SHA1 and HMAC-SHA2-256.</td></tr>
<tr><td>FCS_COP.1/KeyedHash</td><td>The TOE supports keyed-hash message authentication in accordance with HMAC-SHA-256, HMAC-SHA-512 with key sizes 256 and 512 bits and message digest sizes 256, 512 bits. The key length, hash function used, block size, and output MAC lengths are identified in the table below.
<table>
<tr><th>Algorithm</th><th>Key Length</th><th>Hash function</th><th>Block Size</th><th>Output MAC length</th></tr>
<tr><td>HMAC-SHA-256</td><td>256 bits</td><td>SHA-256</td><td>512 bits</td><td>256 bits</td></tr>
<tr><td>HMAC-SHA-512</td><td>512 bits</td><td>SHA-512</td><td>1024 bits</td><td>512 bits</td></tr>
</table></td></tr>
<tr><td>FCS_COP.1/CMAC</td><td>The TOE performs keyed-hash message authentication in accordance with AES-CMAC. The cryptographic key sizes are 128 and 256 bits. The hash function used is AES. The block size is 128 bits. The message digest size (output MAC length) is 128 bits for MACsec cryptographic operations.</td></tr>
<tr><td>FCS_COP.1/MACSEC</td><td>The TOE performs key wrap encryption and decryption in accordance with the AES Key Wrap mechanism specified in NIST SP 800-38F, using AES with key sizes of 128 and 256 bits. The TOE also performs data encryption and decryption using AES in Galois/Counter Mode (GCM) as specified in ISO 19772, with key sizes of 128 and 256 bits.</td></tr>
<tr><td>FCS_RBG_EXT.1</td><td>All random number generation by the TOE is performed in accordance with ISO/IEC 18031:2011 using HMAC_DRBG with ‘HMAC-SHA-512’ implemented in the kernel library. The HMAC_DRBG algorithm is seeded using an software-based entropy source implementing in accordance with NIST Special Publication SP 800-90B</td></tr>
</table>

Requirement	TSS Description
	containing a minimum of 256 bits of entropy. The appliance is to be operated with FIPS mode enabled.
FCS_SSH_EXT.1, FCS_SSHS_EXT.1	The Junos OS SSH Server also supports Trusted Paths using SSHv2 protocol which ensures the confidentiality and integrity of user sessions. The encrypted communication path between Junos OS SSH Server and a remote administrator is provided by the use of an SSH session. Remote administrators of Junos OS initiate communication to the Junos CLI through the SSH tunnel created by the SSH session. Assured identification of Junos OS is guaranteed by using public key-based authentication for SSH. The SSHv2 protocol ensures that the data transmitted over an SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module. The TOE implements SSH server for remote administration that complies with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5656,6668, 8308 and 8332. The TOE can be configured to accept public-key based authentication and/or password-based authentication. The TOE supports keyboard-interactive authentication also. Providing multifactor authentication mechanism would require the use of an external AAA server, which is outside the CC scope, as a result of which the keyboard-interactive authentication method works similarly to the password-based method in the evaluated configuration. The TOE does not require multiple authentication mechanisms for users. The TOE implements key derivation in accordance with RFC 5656 (Section 4), using SHA-256, SHA-384, or SHA-512 (matching the negotiated hash algorithm) as the underlying hash function for deriving encryption keys, MAC keys, and initialization vectors for both client-to-server and server-to-client communications. “Large packets” are defined as the packets that are greater than 256K bytes in an SSH transport connection. SSH packets greater than 256k bytes, when detected by the TOE are dropped and the connection is terminated. The TOE supports the following algorithms for its SSH implementations: • AES-CBC and AES-CTR encryption algorithms with key sizes of 128 bits and 256 bits. The TOE does not support the “none” cipher. • SHA1, SHA2-256, SHA2-384, and SHA2-512 algorithms for hashing. • HMAC-SHA-256 and HMAC-SHA-512 for keyed-hashing. The TOE supports keys generated in accordance with “ssh-rsa”, “rsa-sha2-256”, “rsa-sha2-512”, “ecdsa-sha2-nistp256”, “ecdsa-sha2-nistp384” or “ecdsa-sha2-nistp521”. Key exchange is performed only using one of the supported key exchange algorithms, which are ordered as follows: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521 (all specified in RFC 5656). SSH rekeying is triggered by exceeding either a time or data threshold. Administrators can configure the specific thresholds that initiate rekeying. The time limit for rekeying can be set from 1 to 1440 minutes (24 hours). The data limit for rekeying can be set from 0.5MB to 4GB.

Requirement	TSS Description
FCS_MACSEC_EXT.1	MACsec is implemented in accordance with IEEE 802.1AE-2018. The MACsec provides connectionless user data confidentiality, frame data integrity, and data origin authenticity. Secure channel is identified by Secure Channel Identifier (SCI) that is comprised of a globally unique MAC address and a Port Identifier, unique within the system that has been allocated that address. SCI (8 octets) is appended to every MKPDU packet and the TOE can be configured to enforce SCI tagging such that packets are rejected if they do not have a valid SCI. The TOE allows only Extended Authentication Protocol over LAN (EAPOL - PAE EtherType 88-8E), MACsec frames (EtherType 88-E5) and MAC control frames (EtherType is 88-08) to bypass through its MACsec interface and rejects all other frame types. Also, a filter in PFE traps the packets to RE with ether type 88-8E.
FCS_MACSEC_EXT.2	Each MACsec Key Agreement protocol data unit (MKPDU) transmitted is integrity protected by a 128-bit Integrity Check value (ICV), generated by AES- CMAC using the Integrity Check value Key (ICK). The ICV Key (ICK) is derived from CAK (using AES_CMAC). The Integrity Check Value (ICV) of MACsec protocol data units (MPDUs) is calculated using the SAK over the destination address, source address, SecTAG, and user data (after encryption, if applicable) and is encoded in the last eight to sixteen octets of theMPDU. The length of the ICV is between 8 and 16 octets, depending on the Cipher Suite. The 64 most significant bits of the 96-bit IV used in generating the ICV are the octets of the SCI, and the 32 least significant bits of the 96-bit IV are the octets of the PN. MACsec allows IPv4/v6 and TCP/UDP headers to be unencrypted while the rest of the frame is encrypted. The offset value for MACsec protected frames are: Offset 0 – Default; Encrypts the entire MPDU payload in the frame Offset 30 – IPv4 & TCP/UDP headers are unencrypted and rest of the payload is encrypted Offset 50 – IPv6 & TCP/UDP headers are unencrypted and rest of the payload is encrypted
FCS_MACSEC_EXT.3	SAK is generated using KDF function AES-CMAC-128 or AES-CMAC-256 based on the cipher suite configured using the following transform function: SAK = KDF (Key, Label, KS-nonce | MI-value list | KN, SAKlength) where, • Key= CAK • Label= “IEEE8021 SAK” • KS-nonce = a nonce of the same size as the required SAK, obtained from an RNG each time an SAK is generated. • MI-valuelist = a concatenation of MI values from all live participants • KN = four octets, the Key Number assigned by the Key Server as part of the KI • SAKlength = two octets representing an integer value (128 for a 128 bit SAK, 256 for a 256 bit SAK) with the most significant octet first.

Requirement	TSS Description
	The strength of the session protected by the CAK is determined by the strength of the underlying AES algorithm and its standard key length. Using AES-128 means a 128-bit key and a 2^{128} key space. Using AES-256 means a 256-bit key and a 2^{256} key space. RNG is used for nonce.
FCS_MACSEC_EXT.4	Each distributed SAK is protected by AES Key Wrap method with Key Encryption Key (KEK) as key input.
FCS_MKA_EXT.1	Each MACsec Key Agreement (MKA) protocol data unit (MKPDU) transmitted is integrity protected by an 128-bit Integrity Check value (ICV), generated by AES-CMAC using the Integrity Check value Key (ICK). The ICV Key (ICK) is derived from CAK (using AES_CMAC). The TOE is compliant with IEEE 802.1X-2020 and 802.1Xbx2014 for MKA. The TOE supports an MKA Lifetime Timeout limit of 6.0 seconds and MKA Hello Time limit of 2 seconds. Group CAK is not supported. Peer to peer is the only method supported by the TOE. The key server distributes a SAK by pairwise CAKs that are PSKs. The TOE's PAE supports the establishment of unique CAs exclusively with individual peer devices. MKA within the TOE is implemented to manage peer-to-peer CAKs and derive SAKs solely for point-to-point secure links. Group CAK functionality, including the distribution of new group SAKs based on group membership changes, is not supported by the TOE. Consequently, any changes in the secure communication landscape must be managed through the configuration and re-establishment of peer-to-peer CAKs. The TSF performs robust validation and processing of MACsec Key Agreement Protocol Data Units (MKPDUs) in strict accordance with IEEE 802.1X-2020. Specifically, the TSF validates MKPDUs consistent with the requirements outlined in IEEE 802.1X-2020 Section 11.11.2 and Section 11.11.4. In order to uphold security and integrity, the TSF automatically discards, without further processing, any MKPDU that fails validation by exhibiting any of the following characteristics: • The destination address of the MKPDU was an individual address. • The MKPDU is less than 32 octets long. • The MKPDU comprises fewer octets than indicated by the Basic Parameter Set body length, as encoded in bits 4 through 1 of octet 3 and bits 8 through 1 of octet 4, plus 16 octets of ICV. • The CAK Name is not recognized. Conversely, valid MKPDUs that successfully pass these rigorous checks are decoded and processed in a manner fully consistent with the specifications of IEEE 802.1X-2020 Section 11.11.4, enabling the secure establishment and maintenance of MACsec Key Agreement associations.
FIA_AFL.1	The TOE supports SSH for remote administrative actions. Each unsuccessful authentication attempt is detected and tracked by the TOE. SSH login retries are enabled after the first failed authentication attempt for a username. The maximum number of authentication attempts permitted before SSH connection termination is configurable within the range of 1 to 10. Exceeding this

Requirement	TSS Description
	threshold results in user account lockout, preventing further SSH login attempts until the administrator-configured lockout period (1 to 43,200 minutes) expires. The locking mechanism can be configured to remain locked until an administrator unlocks the account, or it can be configured to unlock after a specified period of time. Even when an account is locked for remote access to the TOE, an administrator is always able to login locally through the serial console.
FIA_PMG_EXT.1	The TOE supports password management capabilities for administrative passwords. Passwords may be composed of any combination of upper and lower-case letters, numbers, specific special characters which include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”,[“~”, “,”, “.”, “/”, “:”, “;”, “_”, “+”, “-“, “=”, “{“, “}”, “[“, “]”, “|”, “<”, “>”. The minimum password length can be configured by the Administrator and can range from 6 to 20 characters.
FIA_UIA_EXT.1	The TOE requires all users to be successfully identified and authenticated as an administrator before allowing any TSF mediated actions to be performed. Access to the TOE is facilitated through directly connecting to the TOE through serial console or remotely connecting to the TOE through SSHv2. Every user that authenticates is logged in with their respective privileges. Regardless of the interface at which the administrator interacts, the TOE prompts the user for a credential. Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality. No TOE administrative access is permitted until an administrator is successfully identified and authenticated. For remote administration, the TOE supports public key authentication and password-based authentication. This login process identifies and authenticates the user using PAM operations. The login process does two things; it first establishes that the requesting user is whom they claim to be and second provides them with an interactive Junos Command interactive command line interface (CLI). If the remote user uses public key-based authentication, the SSH daemon would look up for the user’s public key in the authorized keys file located in the directory ‘.ssh’ in the user’s home directory (i.e. ‘~/.ssh/’) and this authentication method will be attempted before any other if the user has a key available. If the key is found in the authorized keys file, the user is granted access to the TOE. If the user uses password-based authentication and they provide valid username and password, then user is granted access to the TOE. If the user enters invalid user credentials, they will not be granted access. For password authentication, login() interacts with a user to request a username and password to establish and verify the user’s identity. The username entered by the administrator at the username prompt is reflected to the screen, but no feedback to screen is provided while the entry made by the administrator at the password prompt until the Enter key is pressed. login() uses PAM Library calls for the actual verification of this data. The password is hashed and compared to the stored value, and success/failure is indicated to login(). PAM is used in the TOE to support authentication management, account management, session management

Requirement	TSS Description
	and password management. Login primarily uses the session management and password management functionality offered by PAM. The Keyboard-Interactive Based authentication for SSH is supported by default and needs no additional configuration apart from a password being configured for the user. Providing multifactor authentication mechanism would require the use of an external AAA server, which is outside the CC scope, as a result of which the keyboard-interactive authentication method works similarly to the password-based method in the evaluated configuration. For local TOE administration through console, administrators presented with password-based authentication are only granted access to the TOE after entering correct user credentials. The only available actions before remote administrator authentication is displaying the warning banner in accordance with FTA_TAB.1 and responding to ICMP echo requests. The only available action before local administrator's action is to display the warning banner
FIA_UAU.7	None.
FIA_PSK_EXT.1	To generate and configure MACsec encryption using bit-based Pre-Shared Keys (PSK), administrators must configure each device on a point-to-point link with a unique identifier called the 'Connectivity Association Key Name' (CKN) and a secret key known as the 'Connectivity Association Key' (CAK). This same CKN and CAK pair must be manually entered on both ends of the link. The devices (TOE and its peer device) then use this shared secret (the CAK, identified by the CKN) to verify each other's identity and establish a secure MACsec connection. The TSF accepts bit-based preshared keys entered as a string of up to 64 hexadecimal characters.
FMT_MOF.1/ManualUpdate	Performing firmware updates on the TOE is restricted to authorized Security Administrators only.
FMT_MOF.1/Functions	The syslog server can be configured by the administrator on the TOE to transmit audit logs securely in real time via Netconf over SSH. Local audit logs reside in /var/log on the filesystem. Log file access, deletion, and archiving are restricted to authenticated Security Administrators via the CLI or direct filesystem access. The TOE manages audit data through an automated rotation system that archives the active log once it reaches a configurable size (up to 1 GB). The system maintains a set number of compressed archives (1–1000), systematically renaming older files and deleting the oldest once the storage limit is reached. To prevent data loss, the TSF monitors storage capacity, alerting administrators at 92% occupancy while utilizing reserved blocks to continue logging. If storage is completely exhausted, the system logs a final "No space left on device" entry and terminates the logging service, though the device itself remains operational. Administrators can configure these storage limits on the TOE. The TOE allows administrators to modify the behaviour of the following administrative functions using CLI commands: • transmission of audit data to an external IT entity – using the 'system services netconf' configuration hierarchy.

Requirement	TSS Description
	• handling of local audit data– using the 'system syslog' configuration hierarchy.
FMT_MOF.1/Services	The Security Administrator has the capability to: • Start/stop and modify the behaviour of the trusted communication channel to external syslog server using the 'system services netconf' hierarchy command. • Start/stop and modify the behaviour of the trusted communication path for remote administrative sessions (SSH) using the 'system services ssh' hierarchy command. Audit functions can be enabled or disabled, and SSH session settings can be customized, all through manual configurations which are documented in the AGD.
FMT_MTD.1/CoreData	The TOE restricts the ability to manage the TOE to Security Administrators. Administrative users are required to login before being provided with access to any administrative functions. Non-security administrators are not allowed to modify any TOE functions. No interface is available to an unauthenticated user except the login prompt. Any commands used to modify TOE functions are not made available to non-administrative users and its attempt to use them will result in an invalid action error. The authentication failure parameters can only be configured by the administrator for remote access to the TOE. The TOE does not support the handling of X.509v3 certificates or implementation of a trust store for authentication purposes.
FMT_MTD.1/CryptoKeys	The Security Administrator can manage the following keys, with the specified operations available: • **SSH Keys (ECDSA, RSA):** Generation, import, modification, and deletion. • **Pre-shared CAKs:** Configuration (which includes modification, and/or deletion). The instructions for performing these operations on the SSH keys and pre-shared CAKs have been documented in the AGD.

FMT_SMF.1	The available management functions are listed below and these can be accessed via the SSH remotely or locally via console:
	• *Ability to administer the TOE remotely;*
	• *Ability to configure the access banner;*
	• *Ability to configure the remote session inactivity time before session termination;*
	• *Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;*
	• [
	  ○ Ability to start and stop services;
	  ○ Ability to modify the behaviour of the transmission of audit data to an external IT entity
	  ○ Ability to configure local audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full, changes to local audit storage size);
	  ○ Ability to manage the cryptographic keys;

Requirement	TSS Description
	○ Ability to manage the cryptographic functionality; ○ Ability to configure thresholds for SSH rekeying; ○ Ability to re-enable an Administrator account; ○ Ability to set the time which is used for time-stamps; ○ Ability to administer the TOE locally; ○ Ability to configure the local session inactivity time before session termination or locking; ○ Ability to configure the authentication failure parameters for FIA_AFL.1; ○ Ability to manage the trusted public keys database]. The local interface of the TOE can be accessed directly through a serial port connection. The presence of a system variable called "SSH_CONNECTION" is used to determine whether the connection to the TOE is local (via serial) or remote (via SSH). This variable is present during SSH connections and absent during console access. The TOE stores audit logs locally in an audit file. The logs are of persistent nature and these local audit logs are stored in /var/log/ in the underlying filesystem. These audit logs are automatically overwritten as per the administrative configurable limits on storage volume. The default maximum size is 1Gb. Only authorized Security Administrators allowed to either read or delete the log files. These log files can be accessed using command-line interface (CLI) or via direct filesystem access, but only after successfully authentication as a Security Administrator.
FMT_SMF.1/MACSEC	The available management functions are listed below and these can be accessed via the SSH remotely or locally via console: • Ability to generate a PSK and install it in the device • Specify the lifetime of a CAK • CLI commands to manage the Key Server to create, delete, and activate MKA participants • Enable, disable, or delete a PSK-based CAK using CLI commands
FMT_SMR.2	The TOE supports a Security Administrator role, administrable locally or remotely, responsible for provisioning user accounts. Only Security Administrator accounts, associated with the "security-admin" login class and its requisite permissions, can manage the TOE. User accounts comprise a username, password, and role (privilege). System access via the console or SSHv2 is granted to the administrator only after successful identification and authentication.
FPT_SKP_EXT.1	The TOE handles zeroization for all CSP, plaintext secret and private cryptographic keys along with respective key storage and protection of non-plaintext key according to Table 18 – Zeroization of Keys and CSP. The TOE does not provide a CLI interface to permit the viewing of keys. Cryptographic keys are protected through the enforcement of kernel-level file access rights, limiting access to the contents of cryptographic key containers to processes with cryptographic rights or shell users with root permission.

Requirement	TSS Description
FPT_APW_EXT.1	The TOE stores all password authentication data in a secure directory that is not accessible to users. Cryptographic keys are protected through the enforcement of kernel-level file access rights. Passwords are obscured from the user from both local and remote CLI interfaces. The TSF protects administrative passwords by ensuring they are never stored in plaintext. When a password is created or updated, the TSF processes it using a Modular Crypt Format (MCF), which incorporates a unique salt and a SHA256 or SHA512 hash algorithms.
FPT_TST_EXT.1, FPT_FLS.1	The TOE runs the following set of self-tests during power on to check the correct operation of the Junos OS firmware: • Power on test – determines the boot-device responds, and performs a memory size check to confirm the amount of available memory. • File integrity test – verifies the integrity of all mounted signed packages, ensuring that system files have not been tampered with. To test the integrity of the firmware, the SHA-256 fingerprints of executables and other immutable files are recalculated and verified using the ECDSA P-256 digital signature against the signed fingerprints contained in the manifest file. • Crypto integrity test – checks integrity of major CSPs, such as SSH hostkeys and various keys. This test can be run on demand. This test calculates the SHA-256 hash of files in the TOE directory and compares each of them against a stored value in a file with the same file name and the changed extension. If any of the calculated hashes does not match, the test fails. This behaviour is the same across all platforms. • Authentication error – verifies that the veriexec file integrity enforcement framework is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real, which triggers an intentional failure and verifies that it gets detected. • Kernel, libmd, OpenSSL, SSH, MACsec – verifies correct output from known answer tests for appropriate algorithms. • Noise Source Health Test – Noise source health tests verify the correct operation of the noise source. Tests include a repetitive count test and an adaptive proportion test. These NIST-approved self-tests cover all cryptographic primitives and security-critical TSF components and are executed before any security function is used, they provide a complete and reliable demonstration that the TSF is operating correctly. If one of the self-tests fail, the device panics and reboots continuously. You can recover the device using USB install.
FPT_TUD_EXT.1	Security Administrators able to query the current version of the TOE firmware using the CLI command “show version”. The TOE does not support delayed activation or partial updates for the TOE. If a new version of the TOE firmware is available an update of the TOE firmware can be initiated. Updates are downloaded and applied manually. The installable firmware package containing the TOE firmware has a digital signature that is checked when the Security Administrator attempts to install the package. The TOE uses Juniper IMA to verify the package integrity. The hash signature is verified using public key during package installation.

Requirement	TSS Description
	If the image fails the signature check, then the image is deleted from the device and no upgrade occurs. Successful update is only initiated when the signature verification of the TOE passes. The TOE does not support automatic updates.
FPT_STM_EXT.1	All the audit events recorded by TOE are timestamped. The clock function of the TOE provides a source of date and time information for the appliance, used in audit timestamps, which is maintained using the hardware Time Stamp Counter as the clock source. Time on the TOE can only be configured manually by the security administrator. The TSF also relies on timestamps for rekey thresholds, inactivity timeouts and authentication failure-based account lockouts.
FPT_CAK_EXT.1	The TOE protects each CAK by employing AES Key Wrap using System Master Password. The TOE stores all this information in a secure directory (config file) that is not readily accessible to administrators. Cryptographic keys are protected through the enforcement of kernel-level file access rights, limiting access to the contents of cryptographic key containers to processes with cryptographic rights or shell users with root permission.
FPT_RPL.1	To protect against replay (within the Control Plane) each participant in the protocol chooses a random 96-bit member identifier (MI) when MKA begins, and this MI is used, together with a 32-bit message number (MN) initialized to 1 and incremented with each MKPDU transmitted. The Data Plane replay functionality ensures that a man-in-the middle cannot replay a snooped packet or reuse packet number. As bounded receive delay functionality is not supported, it is necessary to configure replay protection in the evaluated configuration using replay-protect. The replay-window-size specifies the number of packets which can be replayed. If set to zero this means no replays are permitted (and should not be used when out of ordering is expected). To prevent replay attacks, the TOE verifies that each incoming MKPDU contains a unique and incremented MN. Any MKPDU featuring a duplicate or outdated MN is discarded and an error message is logged immediately by the TSF.
FTA_SSL.3, FTA_SSL.4, FTA_SSL_EXT.1	User sessions, both remote and local, with the TOE can be terminated by users. The administrative user can logout from the TOE using CLI by typing “exit” or “quit” to terminate the session. The TOE makes the current contents unreadable after the admin initiates termination. No user activity can take place until the user re-identifies and authenticates. User sessions, both remote and local, with the TOE can also be terminated due to inactivity (time threshold). Security Administrators can configure this inactivity timer on user sessions. This setting is controlled by the idle-timeout parameter, which ranges from 0 to 4,294,967,295 minutes. Administrators can configure this inactivity timer via the CLI. Once the set inactivity period is crossed the user would be logged out from the TOE and the session would be terminated. For each user session, the TOE maintains a count of clock cycles (provided by the system clock) since the last user activity. The count is reset each time the user performs any activity within the session. When the counter reaches the number of clock cycles equating to the configured period of inactivity, the user session is terminated.

Requirement	TSS Description
FTA_TAB.1	The TOE can be accessed for administrative purposes either locally via console or remotely using SSH (CLI). For both local and remote methods of access, the TOE provides an access banner prior to login, with the authentication prompt. These banners can be configured by the security administrator for both local and remote login methods. The banner can provide warnings against unauthorized access to the secure switch as well as any other information that the Security Administrator wishes to communicate.
FTP_ITC.1, FTP_ITC.1/MACSEC	Junos OS provides an SSH server to support Trusted Channels using SSHv2 protocol which ensures the confidentiality and integrity of communication with the remote audit server. Export of audit information to a secure, remote server is achieved by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server. The remote audit server initiates the connection. The SSHv2 protocol ensures that the data transmitted over an SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module. Additionally, the TOE implements the MACsec protocol for the protection of layer 2 communications between itself and authenticated MACsec peers. The TOE provides trusted channels using SSHv2 protocol which ensures the confidentiality and integrity of communication with the remote audit server (syslog server). The TOE ensures the non-TSF identity by only allowing administrators to configure the identity information. Export of audit information to a secure, remote server is achieved by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server. The SSHv2 protocol ensures that the data transmitted over a SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module.
FTP_TRP.1/Admin	The TOE supports remote administration using SSH. A trusted path is provided by the TOE using SSHv2 protocol between itself and remote administrators so that the contents of administrative sessions are protected against unauthorized disclosure or modification. Remote administrators can initiate communication to the TOE's CLI through the SSH tunnel created by the SSH session. Assured identification is guaranteed by using password or public key-based authentication mechanisms. The SSHv2 protocol ensures that the data transmitted over a SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module.

6.1 CAVP ALGORITHM CERTIFICATE DETAILS

Each of these cryptographic algorithms have been validated as identified in the table below.

Table 17 – CAVP Algorithm Certificate References

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #	TOE
FCS_CKM.1	RSA schemes using cryptographic key sizes of [**2048, 3072, 4096 bits**] that meet the following: FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;	Junos 23.4R Openssl	RSA KeyGen (FIPS186-5) Modulo: 2048, 3072, 4096	A7286	Intel Xeon D-1627 (Hewitt Lake-DE) AES ECB 128bit & 256bit Encryption/Decryption Engine (BCM82391)
	ECC schemes using 'NIST curves' [**P-256, P-384, P-521**] that meet the following: FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2;	Junos 23.4R Openssl	ECDSA KeyGen (FIPS186-5) Curve: P-256, P-384, P-521 ECDSA KeyVer (FIPS186-5) Curve: P-256, P-384, P-521	A7286	
FCS_CKM.2	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment	Junos 23.4R Openssl	KAS-ECC-SSC Sp800-56Ar3 Domain Parameter Generation Methods:	A7286	

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #	TOE
	Schemes Using Discrete Logarithm Cryptography"		P-256, P-384, P-521		
FCS_COP.1/ DataEncryption	AES used in [**CBC, CTR, GCM**] mode and cryptographic key sizes [**128 bits, 256 bits**] [CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772].	Junos 23.4R Openssl	AES-CBC Direction: Decrypt, Encrypt Key Length: 128, 256 AES-CTR Direction: Decrypt, Encrypt Key Length: 128, 256	A7286	
		AES ECB 128bit & 256bit Encryption/Decryption Engine	AES-GCM Direction: Decrypt, Encrypt Key Length: 128, 256	AES 4545	
FCS_COP.1/ SigGen	For RSA schemes: FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5;	Junos 23.4R Openssl	RSA SigGen (FIPS186-5) Modulo: 2048, 3072, 4096 RSA SigVer (FIPS186-5)	A7286	

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #	TOE
			Modulo: 2048, 3072, 4096		
	For ECDSA schemes implementing [**P-256, P-384, P-521**] curves that meet the following : FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves;	Junos 23.4R Openssl	ECDSA SigGen (FIPS186-5) Curve: P-256, P-384, P-521 ECDSA SigVer (FIPS186-5) Curve: P-256, P-384, P-521	A7286	
FCS_COP.1/ Hash	[**SHA-1, SHA-256, SHA-384, SHA-512**] and message digest sizes [**160, 256, 384, 512**] bits that meet the following: ISO/IEC 10118-3:2004.	Junos 23.4R Openssl	SHA-1 Message Length: 0-65536 Increment 8 SHA2-256 Message Length: 0-65536 Increment 8 SHA2-384 Message Length: 0-65536	A7286	

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #	TOE
			Increment 8 SHA2-512 Message Length: 0-65536 Increment 8		
		Junos 23.4R LibMD	SHA2-256 Message Length: 0-65536 Increment 8 SHA2-512 Message Length: 0-65536 Increment 8	A6550	
FCS_COP.1/ KeyedHash	[**HMAC-SHA- 256, HMAC-SHA-512**] and cryptographic key sizes [**key size (in bits) used in HMAC**] and message digest sizes [**256, 512**] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.	Junos 23.4R Openssl	HMAC-SHA2-256 MAC: 256 Key Length: 256 HMAC-SHA2-512 MAC: 512 Key Length: 512	A7286	

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #	TOE
FCS_RBG_EXT.1	**HMAC_DRBG [SHA-512]** in accordance with ISO/IEC 18031:2011	Junos 23.4R Kernel	HMAC DRBG Mode: SHA2-512	A6620	
FCS_COP.1.1/CMAC	cryptographic algorithm [*AES-CMAC*] and cryptographic key sizes **[*128, 256*] bits and message digest size of 128 bits** that meets the following: [*NIST SP 800-38B*]	Junos 23.4R Macsec	AES-CMAC Key Length: 128, 256	A6551	
FCS_COP.1.1/MACSEC	cryptographic algorithm [*AES used in AES Key Wrap, GCM*] and cryptographic key sizes **[*128, 256*] bits** that meets the following: [*AES as specified in ISO 18033-3, AES Key Wrap as specified in NIST SP 800-38F, GCM as specified in ISO 19772*].	Junos 23.4R Macsec	AES-KW Key Length: 128, 256	A6551	
		AES ECB 128bit & 256bit Encryption/Decryption Engine	AES-GCM Key Length: 128, 256	AES 4545	

6.2 CRYPTOGRAPHIC KEY DESTRUCTION

The table below describes the key zeroization provided by the TOE and as referenced in FCS_CKM.4.

Table 18 – Zeroization of Keys and CSP

Keys/CSPs	Purpose	Method of storage	Storage Location	Method of Zeroization
SSH Private Host Key	The first time SSH is configured, the key is generated. Used to identify the host.	Plaintext	File format on SDD	When the appliance is recommissioned, the config files (including CSP files such as SSH keys) are removed using the "request vmhost zeroize no-forwarding" option.
	Loaded into memory to complete session establishment	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
SSH Session Key	Session keys used with SSH, AES 128, 256, hmac-sha-1, hmac-sha2-256 or hmacsha2-512 key (160, 256 or 512), DH Private Key (2048 or elliptic curve 256/384/521-bits)	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
User Password	Plaintext value as entered by user	Plaintext as entered	Processed in Memory	Memory free() operation is performed by Junos upon completion of authentication
		Hashed when stored (SHA256 and SHA512)	Stored on disk	When the appliance is recommissioned, the config files (including the obfuscated password) are removed using the "request vmhost zeroize no-forwarding" option.
RNG State	Internal state and seed key of RNG	Plaintext	Memory	Handled by kernel, overwritten with zero's at reboot.
MACsec CAK	Pre-shared, static Connectivity Association Key	Encrypted using GCM-AES-256 using System Master Password	Stored in config file	Actively zeroized using "request vmhost zeroize noforwarding

Keys/CSPs	Purpose	Method of storage	Storage Location	Method of Zeroization
MACsec SAK	Security Association Key	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
MACsec KEK	Key Encryption Key	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
MACsec ICK	Integrity Check Key	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
System Master Password	Password used to derive encryption key used for protecting MACsec CAK	Plaintext	disk	Actively zeroized using "request vmhost zeroize noforwarding"

7. ACRONYM TABLE

Acronyms should be included as an Appendix in each document.

Table 19 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CC	Common Criteria
CRL	Certificate Revocation List
DTLS	Datagram Transport Layer Security
EP	Extended Package
GUI	Graphical User Interface
IP	Internet Protocol
NDcPP	Network Device Collaborative Protection Profile
NIAP	Nation Information Assurance Partnership
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
PP	Protection Profile
RSA	Rivest, Shamir & Adleman
SFR	Security Functional Requirement
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSS	TOE Summary Specification
ICV	Integrity Check Value
MKPDU	MACsec Key Protocol Data Unit
TSF	TOE Security Functions
IEEE	Institute of Electrical and Electronics Engineers
CAK	Connectivity Association Key
MKA	MACsec Key Agreement
PSK	Pre-Shared Key
SAK	Secure Association Key
ICK	Integrity Check Key
KDF	Key Derivation Function
MAC	Media Access Control
CKN	Connectivity Association Key Name
NIST	National Institute of Standards and Technology

Acronym	Definition
CAVP	Cryptographic Algorithm Validation Program
VLAN	Virtual Local Area Network
MPD	MACsec Protocol Data
MPDU	MACsec Protocol Data Unit
SCI	Secure Channel Identifier
EAPOL	Extensible Authentication Protocol over LAN
GCM	Galois/Counter Mode
ISO	International Organization for Standardization
PFE	Packet Forwarding Engine
RE	Routing Engine
PoE	Power over Ethernet
LLDP	Link Layer Discovery Protocol
DHCP	Dynamic Host Configuration Protocol
ARP	Address Resolution Protocol
STP	Spanning Tree Protocol
EAP	Extensible Authentication Protocol
LAN	Local Area Network
SA	Security Association
Tx	Transmit
Rx	Receive