Junos OS 24.2R2 for EX4400

Security Target

Version 1.1

May 2026

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Description
1.0	30 MAR 2026	Release for Check Out
1.1	27 MAY 2026	Address ECR comments

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the Junos OS 24.2R2 for EX4400 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 The Juniper Networks EX4400 series comprises high-performance Ethernet switches that support MACsec encryption and offer Gigabit/10-gigabit Ethernet connectivity.

3 The TOE is the Juniper Networks EX4400-24X, EX4400-24T, EX4400-24P, EX4400-24MP, EX4400-48T, EX4400-48P, EX4400-48F, EX4400-48MP platforms executing the Junos OS 24.2R2 software.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	Junos OS 24.2R2 for EX4400 Build: 24.2R2-S4.5
Security Target	Junos OS 24.2R2 for EX4400 Security Target, v1.1

1.3 Conformance Claims

4 This ST supports the following conformance claims:

- CC version 3.1 Revision 5:
  - i) CC Part 2 extended
  - ii) CC Part 3 conformant
- PP-Configuration for Network Devices and MACsec Ethernet Encryption, Version 2.0 (CFG_NDcPP-MACsec_V2.0):

  Base PP: collaborative Protection Profile for Network Devices, Version 3.0e (CPP_ND_V3.0E)

  PP-Module: PP-Module for MACsec Ethernet Encryption, Version 1.0 (MOD_MACSEC_V1.0)
- Functional Package for Secure Shell (SSH), Version 1.0 (PKG_SSH_V1.0) conformant
- NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	Name	Source	Applicability
TD0990	NIT Technical Decision: CTR_DRBG in FCS_RBG_EXT.1.2	CPP_ND_V3.0E	Applicable
TD0973	NIT Technical Decision: FCS_(D)TLSS_EXT.1.3	CPP_ND_V3.0E	Not Applicable.

TD #	Name	Source	Applicability
	Test 2 DHE Ciphersuite Conditionality		TOE does not claim TLS.
TD0967	Allowance of Kex-strict in PKG_SSH_V1.0	PKG_SSH_V1.0	Applicable
TD0939	Updated Conformance Claims for MOD_MACSEC	MOD_MACSEC_V1.0	Applicable
TD0923	NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2	CPP_ND_V3.0E	Applicable
TD0921	NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment	CPP_ND_V3.0E	Applicable
TD0909	Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	PKG_SSH_V1.0	Applicable
TD0900	NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	CPP_ND_V3.0E	Applicable
TD0899	NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2	CPP_ND_V3.0E	Not Applicable. TOE does not claim TLS.
TD0891	Correlation of Implicitly Satisfied Requirements when CPP_ND_V3.0E is the Base-PP	MOD_MACSEC_V1.0	Applicable
TD0889	Correction For Tests Incorrectly Requiring Group MACsec	MOD_MACSEC_V1.0	Applicable
TD0886	Clarification to FAU_STG_EXT.1 Test 6	CPP_ND_V3.0E	Applicable
TD0884	Expansion of Permitted EtherTypes in FCS_MACSEC_EXT.1.4	MOD_MACSEC_V1.0	Applicable
TD0882	MACsec Data Delay Protection, Key Agreement, and Conditional Support for Group CAK	MOD_MACSEC_V1.0	Applicable

TD #	Name	Source	Applicability
TD0881	Correction to MN Usage for FPT_RPL.1 Test	MOD_MACSEC_V1.0	Applicable
TD0880	NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1	CPP_ND_V3.0E	Applicable
TD0879	NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E	CPP_ND_V3.0E	Applicable
TD0870	Security Objectives Rationale for MOD_MACSEC_V1.0	MOD_MACSEC_V1.0	Applicable
TD0868	NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	CPP_ND_V3.0E	Not Applicable. TOE does not claim IPsec.
TD0840	Alignment of Test 22.1 to FMT_SMF.1/MACSEC	MOD_MACSEC_V1.0	Applicable
TD0836	NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1	CPP_ND_V3.0E	Applicable
TD0826	Aligning MOD_MACSEC_V1.0 with CPP_ND_V3.0E	MOD_MACSEC_V1.0	Applicable
TD0825	Correction to IEEE 802.1X Reference	MOD_MACSEC_V1.0	Applicable
TD0816	Clarity for MACsec Self Test Failure Response	MOD_MACSEC_V1.0	Applicable
TD0803	Clarification for Configurable MACsec CKN Length	MOD_MACSEC_V1.0	Applicable
TD0777	Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	PKG_SSH_V1.0	Applicable
TD0746	Correction to FPT_RPL.1 Test 25	MOD_MACSEC_V1.0	Applicable
TD0732	FCS_SSHS_EXT.1.3 Test 2 Update	PKG_SSH_V1.0	Applicable

TD #	Name	Source	Applicability
TD0728	Corrections to MACSec PP-Module SD	MOD_MACSEC_V1.0	Applicable
TD0695	Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.	PKG_SSH_V1.0	Applicable
TD0682	Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	PKG_SSH_V1.0	Applicable

1.4 Terminology

Table 3: Terminology

Term	Definition
CA	MACsec Connectivity Association
CAK	Connectivity Association Key
CC	Common Criteria
CKN	Connectivity Association Key Names
CLI	Command Line Interface
ICK	Integrity Check Value Key
ICV	Integrity Check Value
IEEE	Institute of Electrical and Electronics Engineers
IV	Initialization Vector
KDF	Key Derivation Function
KN	Key Number
KW	Key Wrap
MI	Member Identifier
MKA	MACsec Key Agreement
MKPDU	MACsec Key Agreement Protocol Data Unit
MN	Message Number
MPDU	MACsec Protocol Data Units

Term	Definition
NDcPP	collaborative Protection Profile for Network Devices
PFE	Packet Forwarding Engine
PAE	Port Access Entity
PP	Protection Profile
PSK	Pre-Shared Key
RE	Routing Engine
SAK	Secure Association Key
SCI	Secure Channel Identifier
TOE	Target of Evaluation
TSF	TOE Security Functionality

2 TOE Description

2.1 Type

5 The TOE is a MACsec capable network device.

2.2 Usage

2.2.1 Deployment

6 Offering a full suite of Layer 2 and Layer 3 capabilities, the EX4400 switches can be deployed in branch and campus access/distribution layer networks or as top-of-rack/core switches in data center environments.

2.2.2 Interfaces

The TOE interfaces are shown in

7 Figure 1.

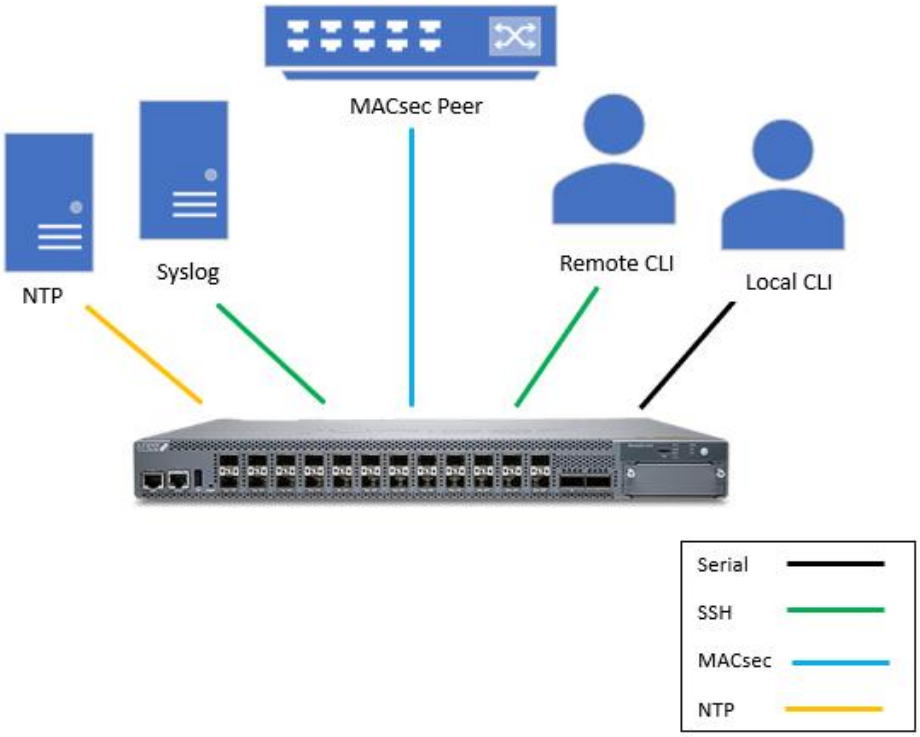

Figure 1: TOE interfaces

8 The TOE interfaces are as follows:

- **Local CLI.** Direct serial connection to the local CLI.
- **Remote CLI.** Remote connection to the CLI via SSHv2.
- **Syslog.** Logging to syslog via SSHv2.
- **MACsec.** Point-to-point connections with MACsec peers.

- **NTP.** The TOE synchronizes time via NTP.

2.3 Security Functions / Logical Scope

9 The TOE provides the following security functions:

- **Security Audit.** The TOE generates audit records of user and administrator actions. The TOE includes the user identity in audit events resulting from actions of identified users. The TOE is capable of sending logs in real-time to an external syslog server via SSHv2.
- **Cryptographic Support** The TOE implements a cryptographic module. The cryptographic module has the ability to generate and destroy cryptographic keys. Relevant Cryptographic Algorithm Validation Program (CAVP) certificates are shown in Table 4.
- **Identification and Authentication.** The TOE ensures that all users must be identified and authenticated before accessing its functions and data.
- **Security Management.** The TOE provides a suite of security management functions that is available to authorized administrators in support of configuring and maintaining the TOE.
- **Protection of the TSF.** The TOE performs a suite of self-tests to ensure the correct operation and enforcement of its security functions including software integrity verification using digital signatures. The TOE also supports the use of NTP or its own manually configurable time source to support the execution of reliable and time sensitive operations.
- **TOE Access.** The TOE can be accessed physically via direct serial connection or remotely via SSH connection. When a user account has reached the configured number of failed authentication attempts, the account will be locked for a Security Administrator defined time period.
- **Trusted Path/Channels.** The TOE protects the integrity and confidentiality of communications between itself and local/remote administrators as noted in section 2.2.2 above.

Table 4: CAVP Certificates

Module		Certificate
Junos 24.2R2 – openSSL (OpenSSL for Junos OS 24.2R2, based on version 1.1.1zb)		A6931
Junos 24.2R2 – libmd (LibMD for Junos OS 24.2R2, created from same sources as OpenSSL version 1.1.1zb)		A6930
Junos 24.2R2 – kernel (Kernel for Junos OS 24.2R2, based on FreeBSD-15 Stable release)		A6929
Junos 24.2R2 – Macsec (MACSEC Library for Junos OS 24.2R2)		A6928
	BCM54192, BCM54195	AES 4544

Module		Certificate
AES ECB 128bit & 256bit Encryption/Decryption Engine (MACsec FPGA(s))	BCM82398, BCM82399	AES 4545
	BCM82756	AES 4550
	BCM84894M	C996
	BCM54998EM	C1869

2.4 Physical Scope

10 The physical boundary of the TOE includes all software and hardware shown in Table 5. The TOE is delivered via commercial courier.

Table 5: TOE Models

Model	Port Configuration	MACsec FPGA	CPU	Software
EX4400-24X	24-Port 1/10GbE SFP+	PHY1 BCM82756 PHY2 BCM82399 PHY3 BCM82398	Intel Atom C3558R (Denverton)	Junos OS 24.2R2
EX4400-24T	24-port 10/100/1000BASE-T	PHY1 BCM54192 PHY2 BCM82399		
EX4400-24P	24-port 10/100/1000BASE-T (PoE)	PHY1 BCM54192 PHY2 BCM82399		
EX4400-24MP	24x-port 100M/1/2.5/5/10GbE (PoE)	PHY1 BCM84894M PHY2 BCM82399		
EX4400-48T	48-port 10/100/1000BASE-T	PHY1 BCM54192 PHY2 BCM82399		

Model	Port Configuration	MACsec FPGA	CPU	Software
EX4400-48P	48-port 10/100/1000BASE-T (PoE)	PHY1 BCM54192 PHY2 BCM82399		
EX4400-48F	12-port 1000/10000BASE-X + 36-port 100/1000BASE-X	PHY1 BCM54195 PHY2 BCM82399 PHY3 BCM82756		
EX4400-48MP	12x100M/1/2.5/5/10GbE + 36x100M/1/2.5GbE (PoE)	PHY1 BCM54998EM PHY2 BCM82399 PHY3 BCM84894M		

2.4.1 Guidance Documents

11 The TOE includes the following guidance documents (PDF):

- [CCGUIDE] - Juniper Networks Junos OS Common Criteria Evaluated Configuration Guide for EX4400-24MP, EX4400-24P, EX4400-24T, EX4400-24X, EX4400-48F, EX4400-48MP, EX4400-48P, and EX4400-48T Devices, Release 24.2R2, Published 2026-03-27

2.4.2 Non-TOE Components

12 The TOE operates with the following components in the environment:

- **Audit Server.** The TOE sends audit events to an external logging server via SSHv2.
- **NTP Server.** The TOE synchronizes time via NTP.
- **MACsec Peer(s).** TOE MACsec peers supporting 802.1ae.

2.4.3 Functions Excluded from the TOE Evaluation

13 The following functions are excluded from the evaluation and should not be used in the evaluated configuration:

- Telnet
- FTP

- SNMP
- TLS/SSL including management via J-Web, JUNOScript and JUNOScope

3 Security Problem Definition

3.1 Threats

Table 6: Threats (CPP_ND_V3.0E)

Identifier	Description
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 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

Identifier	Description
	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.

Table 7: Threats (MOD_MACSEC_V1.0)

Identifier	Description
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 sit on the periphery of a network, which means that it may have an externally- facing 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

Identifier	Description
	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

Table 8: Assumptions (CPP_ND_V3.0E)

Identifier	Description
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.
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). If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform.
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).

Identifier	Description
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. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
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.

14 **NOTE:** MOD_MACSEC_V1.0 does not define additional assumptions.

3.3 Organizational Security Policies

Table 9: Organizational Security Policies (CPP_ND_V3.0E)

Identifier	Description
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.

15 **NOTE:** MOD_MACSEC_V1.0 does not define additional OSPs.

4 Security Objectives

16 **NOTE:** CPP_ND_V3.0E does not define any security objectives for the TOE.

Table 10: Security Objectives for the TOE (MOD_MACSEC_V1.0)

Identifier	Description
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. Addressed by: FCS_MACSEC_EXT.4, FCS_MKA_EXT.1, FIA_PSK_EXT.1, FCS_DEVID_EXT.1 (selection-based), FCS_EAP-TLS_EXT.1 (selection- based)
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. Addressed by: FMT_SMF.1/MACSEC, FPT_CAK_EXT.1, FIA_AFL_EXT.1 (optional), FTP_TRP.1/MACSEC (optional), FMT_SNMP_EXT.1 (selection-based)
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. Addressed by: FCS_COP.1/CMAC, FCS_COP.1/MACSEC, FCS_MACSEC_EXT.2, FCS_MACSEC_EXT.3, FTP_ITC.1/MACSEC, FTP_TRP.1/MACSEC (optional), FCS_SNMP_EXT.1 (selection-based)
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). Addressed by: FCS_MACSEC_EXT.1, FIA_PSK_EXT.1, FPT_DDP_EXT.1

Identifier	Description
O.REPLAY_ DETECTION	A MACsec device is expected to help mitigate the threat of MACsec data integrity violations by providing a mechanism to detect and discard replayed traffic for MPDUs. Addressed by: FPT_RPL.1, FPT_RPL_EXT.1 (optional)
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). Addressed by: FAU_GEN.1/MACSEC
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. Addressed by: FPT_FLS.1

Table 11: Security Objectives for the Operational Environment (CPP_ND_V3.0E)

Identifier	Description
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 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.
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.

Identifier	Description
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.

17 **NOTE:** MOD_MACSEC_V1.0 does not define additional security objectives for the operational environment.

5 Security Requirements

5.1 Conventions

18 This document uses the following font conventions to identify the operations defined by the CC:

- **Assignment.** Indicated with italicized text.
- **Refinement.** Indicated with bold text and strikethroughs.
- **Selection.** Indicated with underlined text.
- **Assignment within a Selection:** Indicated with italicized and underlined text.
- **Iteration.** Indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").

19 **Note:** Operations performed within the Security Target are denoted within brackets "[ ]". Operations shown without brackets are reproduced from the Protection Profiles.

5.2 Extended Components Definition

20 The Extended Components are defined in Appendix C of the CPP_ND_V3.0E and MOD_MACSEC_V1.0.

5.3 Security Functional Requirements

Table 12: Summary SFRs

SFR	Title	Source	Requirement
Collaborative Protection Profile for Network Devices, Version 3.0e			
FAU_GEN.1	Audit Data Generation	CPP_ND_V3.0E	Mandatory
FAU_GEN.2	User Identity Association	CPP_ND_V3.0E	Mandatory
FAU_STG_EXT.1	Protected Audit Event Storage	CPP_ND_V3.0E	Mandatory
FAU_STG.1	Protected Audit Trail storage	CPP_ND_V3.0E	Optional
FCS_CKM.1	Cryptographic Key Generation	CPP_ND_V3.0E	Mandatory
FCS_CKM.2	Cryptographic Key Establishment	CPP_ND_V3.0E	Mandatory
FCS_CKM.4	Cryptographic Key Destruction	CPP_ND_V3.0E	Mandatory
FCS_COP.1 /DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)	CPP_ND_V3.0E	Mandatory

SFR	Title	Source	Requirement
FCS_COP.1 /SigGen	Cryptographic Operation (Signature Generation and Verification)	CPP_ND_V3.0E	Mandatory
FCS_COP.1 /Hash	Cryptographic Operation (Hash Algorithm)	CPP_ND_V3.0E	Mandatory
FCS_COP.1 /KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)	CPP_ND_V3.0E	Mandatory
FCS_NTP_EXT.1	NTP Protocol	CPP_ND_V3.0E	Selection
FCS_RBG_EXT.1	Random Bit Generation	CPP_ND_V3.0E	Mandatory
FIA_AFL.1	Authentication Failure Handling	CPP_ND_V3.0E	Selection
FIA_PMG_EXT.1	Password Management	CPP_ND_V3.0E	Selection
FIA_UIA_EXT.1	User Identification and Authentication	CPP_ND_V3.0E	Mandatory
FIA_UAU.7	Protected Authentication Feedback	CPP_ND_V3.0E	Selection
FMT_MOF.1 /ManualUpdate	Management of Security Functions Behaviour	CPP_ND_V3.0E	Mandatory
FMT_MOF.1 /Services	Management of Security Functions Behaviour	CPP_ND_V3.0E	Selection
FMT_MOF.1 /Functions	Management of Security Functions Behaviour	CPP_ND_V3.0E	Selection
FMT_MTD.1 /CoreData	Management of TSF Data	CPP_ND_V3.0E	Mandatory
FMT_MTD.1 /CryptoKeys	Management of TSF Data	CPP_ND_V3.0E	Selection
FMT_SMF.1	Specification of Management Functions	CPP_ND_V3.0E	Mandatory
FMT_SMR.2	Restrictions on Security Roles	CPP_ND_V3.0E	Mandatory
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)	CPP_ND_V3.0E	Mandatory

SFR	Title	Source	Requirement
FPT_APW_EXT.1	Protection of Administrator Passwords	CPP_ND_V3.0E	Selection
FPT_STM_EXT.1	Reliable Time Stamps	CPP_ND_V3.0E	Mandatory
FPT_TST_EXT.1	TSF Testing	CPP_ND_V3.0E	Mandatory
FPT_TUD_EXT.1	Trusted Update	CPP_ND_V3.0E	Mandatory
FTA_SSL_EXT.1	TSF-initiated Session Locking	CPP_ND_V3.0E	Selection
FTA_SSL.3	TSF-initiated Termination	CPP_ND_V3.0E	Mandatory
FTA_SSL.4	User-initiated Termination	CPP_ND_V3.0E	Mandatory
FTA_TAB.1	Default TOE Access Banners	CPP_ND_V3.0E	Mandatory
FTP_ITC.1	Inter-TSF trusted channel	CPP_ND_V3.0E	Mandatory
FTP_TRP.1 /Admin	Trusted Path	CPP_ND_V3.0E	Mandatory
Functional Package for Secure Shell (SSH), Version 1.0			
FCS_SSH_EXT.1	SSH Protocol	PKG_SSH_V1.0	Mandatory
FCS_SSHS_EXT.1	SSH Protocol – Server	PKG_SSH_V1.0	Selection
PP-Module for MACsec Ethernet Encryption, Version 1.0			
FAU_GEN.1 /MACSEC	Audit Data Generation (MACsec)	MOD_MACSEC_V1.0	Mandatory
FCS_COP.1 /CMAC	Cryptographic Operation (AES-CMAC Keyed Hash Algorithm)	MOD_MACSEC_V1.0	Mandatory
FCS_COP.1 /MACSEC	Cryptographic Operation (MACsec AES Data Encryption and Decryption)	MOD_MACSEC_V1.0	Mandatory
FCS_MACSEC_EXT.1	MACsec	MOD_MACSEC_V1.0	Mandatory
FCS_MACSEC_EXT.2	MACsec Integrity and Confidentiality	MOD_MACSEC_V1.0	Mandatory
FCS_MACSEC_EXT.3	MACsec Randomness	MOD_MACSEC_V1.0	Mandatory
FCS_MACSEC_EXT.4	MACsec Key Usage	MOD_MACSEC_V1.0	Mandatory

SFR	Title	Source	Requirement
FCS_MKA_EXT.1	MACsec Key Agreement	MOD_MACSEC_V1.0	Mandatory
FIA_PSK_EXT.1	Pre-Shared Key Composition	MOD_MACSEC_V1.0	Mandatory
FMT_SMF.1 /MACSEC	Specification of Management Functions (MACsec)	MOD_MACSEC_V1.0	Mandatory
FPT_CAK_EXT.1	Protection of CAK Data	MOD_MACSEC_V1.0	Mandatory
FPT_FLS.1	Failure with Preservation of Secure State	MOD_MACSEC_V1.0	Mandatory
FPT_RPL.1	Replay Detection	MOD_MACSEC_V1.0	Mandatory
FTP_ITC.1 /MACSEC	Inter-TSF Trusted Channel (MACsec Communications)	MOD_MACSEC_V1.0	Mandatory

5.3.1 Security Audit (FAU)

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 2**~~ **Table 13**.*

Table 13: Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
FAU_STG.1	None.	None.
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_NTP_EXT.1	• Configuration of a new time server • Removal of configured time server	Identity if new/removed time server
FCS_RBG_EXT.1	None.	None.
FCS_SSH_EXT.1	• [Failure to establish SSH connection] • [Establishment of SSH connection] • [Termination of SSH connection session] • [Dropping of packet(s) outside defined size limits]	• [Reason for failure and Non-TOE endpoint of attempted connection (IP Address)] • [Non-TOE endpoint of connection (IP Address)] • [Non-TOE endpoint of connection (IP Address)] • [Packet size]
FCS_SSHS_EXT.1	No events specified	
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_UIA_EXT.1	All use of identification and authentication mechanisms.	Origin of the attempt (e.g., IP address).

Requirement	Auditable Events	Additional Audit Record Contents
FIA_UAU.7	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Services	None.	None.
FMT_MOF.1/Functions	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. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	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_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session lock	None.
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_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: This table has been modified by TD0777.

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 2~~ Table 13***.

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.

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*].

FAU_STG_EXT.1.5 The TSF shall [overwrite previous audit records according to the following rule: [*oldest log 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 [ability to view locally].

FAU_STG.1 Protected Audit Trail Storage

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

5.3.2 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

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

- RSA schemes using cryptographic key sizes of [*2048 bits, 3072 bits, 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].~~

Application Note: This SFR has been modified by TD0921.

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*].~~

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***.

FCS_COP.1/DataEncryption Cryptographic Operation (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] *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].

FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

FCS_COP.1.1/SigGen 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 bits, 3072 bits, 4096 bits*],
- For ECDSA: [*256 bits, 384 bits, 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.

].

Application Note: This SFR has been modified by TD0921.

FCS_COP.1/Hash Cryptographic Operation (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*.

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 bits, 512 bits]* **and message digest sizes [256, 512] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

FCS_NTP_EXT.1 NTP Protocol

FCS_NTP_EXT.1.1 The TSF shall use only the following NTP version(s) [NTP v4 (RFC 5905)].

FCS_NTP_EXT.1.2 The TSF shall update its system time using [

- Authentication using [SHA256] as the message digest algorithm(s);

].

FCS_NTP_EXT.1.3 The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4 The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

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 [[*4*] software-based noise source, [*1*] platform-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",

].

Application Note: This SFR has been modified by TD0990.

5.3.3 Identification and Authentication (FIA)

FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*2 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 Administrator defined time period has elapsed].

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: ["!", "@", "#", "$", "%", "^", "&", "*", "(", ")", *[and all other printable standard ASCII, extended ASCII, and Unicode characters]*];

b. Minimum password length shall be *configurable to between [10] and [20] characters.*

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;
- [[*Negotiation of SSH session, ICMP echo*]]

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 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.

Application Note: This SFR has been modified by TD0900.

FIA_UAU.7 Protected Authentication Feedback

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**.

5.3.4 Security Management (FMT)

FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

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

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*.

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*.

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*.

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*.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 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 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 modify the behaviour of the transmission of audit data to an external IT entity;
- Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;
- Ability to manage the cryptographic keys;
- Ability to configure the cryptographic functionality;
- Ability to configure thresholds for SSH rekeying;
- Ability to set the time which is used for time-stamps;
- Ability to configure NTP;
- 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;

].

Application Note: This SFR has been modified by TD0880.

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.

5.3.5 Protection of the TSF (FPT)

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.

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.

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, synchronise time with an NTP server].

FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1 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 [at no other time] to verify correct operation of cryptographic implementation necessary to fulfil the TSF;
- [start-up] self-tests [*Power On Test, Crypto Integrity Test, Authentication Error, Known Answer Tests (KAT)*].

to demonstrate the correct operation of the TSF.

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

Application Note: This SFR has been modified by TD0836.

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.

5.3.6 TOE Access (FTA)

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.

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*.

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.

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.

5.3.7 Trusted path/channels (FTP)

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***].

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*.

5.4 PKG_SSH Functional Requirements

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1 The TOE shall implement *SSH* acting as a [server] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [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),
- "publickey" (RFC 4252): [
    - 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 [*256KB*] 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.

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),

].

5.5 MOD_MACSEC Functional Requirements

5.5.1 Security Audit (FAU)

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 2)~~ Table 14*]

Table 14: MOD_MACSEC Auditable Events

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 2)~~ Table 14*].

5.5.2 Cryptographic Support (FCS)

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*].

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*].

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 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.

Application Note: This SFR has been modified by TD0884.

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.

FCS_MACSEC_EXT.3 MACsec Randomness

FCS_MACSEC_EXT.3.1 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.

Application Note: This SFR has been modified by TD0825.

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 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).

Application Note: This SFR has been modified by TD0825.

FCS_MKA_EXT.1 MACsec Key Agreement

FCS_MKA_EXT.1.1 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 The TSF shall enforce an MKA Lifetime Timeout limit of 6.0 seconds and [MKA Hello Time limit of 2 seconds, MKA Bounded Hello Time limit of 0.5 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 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.

Application Note: This SFR has been modified by TD0882 and TD0825.

5.5.3 Identification and Authentication (FIA)

FIA_PSK_EXT.1 Pre-Shared Key Composition

FIA_PSK_EXT.1.1 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.

Application Note: This SFR has been modified by TD0825.

5.5.4 Security Management (FMT)

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* [[*using CLI management commands*]]
- *Specify the lifetime of a CAK*
- *Enable, disable, or delete a PSK-based CAK using [[CLI management commands]]*

[

- No other MACsec management functions

]].

5.5.5 Protection of the TSF (FPT)

FPT_CAK_EXT.1 Protection of CAK Data

FPT_CAK_EXT.1.1 The TSF shall prevent reading of CAK values by administrators.

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 noise source health tests*].

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.

5.5.6 Trusted Path/Channels (FTP)

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*].

5.6 Assurance Requirements

21 The TOE security assurance requirements are reproduced from CPP_ND_V3.0E and are summarized in Table 15.

Table 15: Security Assurance Requirements

Assurance Class	Assurance Components
Security Target (ASE)	Conformance Claims (ASE_CCL.1)
	Extended Components Definition (ASE_ECD.1)
	ST Introduction (ASE_INT.1)
	Security Objectives for the Operational Environment (ASE_OBJ.1)
	Stated Security Requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE Summary Specification (ASE_TSS.1)
Development (ADV)	Basic Functional Specification (ADV_FSP.1)
Guidance Documents (AGD)	Operational User Guidance (AGD_OPE.1)
	Preparative Procedures (AGD_PRE.1)
Life Cycle Support (ALC)	Labelling of the TOE (ALC_CMC.1)
	TOE CM Coverage (ALC_CMS.1)
	Systematic Flaw Remediation (ALC_FLR.3)
Tests (ATE)	Independent Testing – Conformance (ATE_IND.1)
Vulnerability Assessment (AVA)	Vulnerability Survey (AVA_VAN.1)

22 In accordance with section 7.1 of the NDcPP, the following refinement is made to ASE:

- **ASE_TSS.1.1C Refinement:** The TOE summary specification shall describe how the TOE meets each SFR. **In the case of entropy analysis, the TSS is used in conjunction with required supplementary information on Entropy.**

6 TOE Summary Specification

23 The following sections describe how the TOE fulfils each SFR.

6.1 Security Audit (FAU)

6.1.1 FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1, FAU_STG.1, FAU_GEN.1/MACSEC

24 The TOE generates and stores audit records for all auditable events. Each event and the content recorded is detailed in Table 13 (***FAU_GEN.1***). Additional auditable events that are generated for MACsec are described in Table 14 (***FAU_GEN.1/MACSEC***). Auditing is implemented using syslog.

25 The detail of what events are to be recorded by syslog are determined by the logging level specified in the "level" argument of the "`set system syslog`" CLI command. To ensure compliance with the requirements the audit knobs detailed in [CCGUIDE] must be configured.

26 At a minimum, Junos OS records with each log entry the date and time of the event, the type of event, subject identity (where applicable) and the outcome (success or failure) of the event (where applicable).

27 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 configured for SSH public key authentication –the hash of the public key that is to be used for authentication is recorded in syslog

28 For SSH (ephemeral) session keys, the PID is used as the key reference to relate the key generation and key destruction audit events. The key destruction event is recorded as a session disconnect event. For example, key generation and key destruction events for a single SSH session key would be reflected by records similar to the following:

```
Sep 27 15:09:36 yeti sshd[6529]: Accepted publickey for root from
10.163.18.165 port 45336 ssh2: RSA
SHA256:l1vri77TPQ4VaupE2NMYiUXPnGkqBWIgD5vW0OuglGI

…

Sep 27 15:09:40 yeti sshd[6529]: Received disconnect from
10.163.18.165 port 45336:11: disconnected by user

Sep 27 15:09:40 yeti sshd[6529]: Disconnected from 10.163.18.165
port 45336
```

29 SSH keys used for trusted channels are NOT deleted by mgd when SSH is de-configured. Hence, the only time SSH keys used for trusted channels are deleted is when a "*request system zeroize*" action is performed and the whole appliance is zeroized (which by definition cannot be recorded).

30 The TOE consists of a single standalone component that stores audit data locally. Syslog can be configured to store the audit logs locally (***FAU_STG_EXT.1***). Audit logs can also be sent to one or more external syslog servers in real time via Netconf over SSH. The transmission of audit logs is done automatically without Administrator intervention (***FAU_STG.1, FMT_MOF.1/Functions***).

31 Local audit log are stored in */var/log/* in the filesystem. Only a Security Administrator can read or delete logs and archive files through the CLI interface or through direct access to the filesystem. The syslogs are automatically deleted locally according to configurable limits on storage volume. The default maximum size is 1Gb. The default maximum size can be modified by the administrator, using the "*size*" argument for the "`set system syslog`" CLI command. The minimum size of a local log file is 64KB.

32 The Junos OS defines an active log file and a number of "archive" files (10 by default, but configurable from 1 to 1000). When the active log file reaches its maximum size, the logging utility closes the file, compresses it, and names the compressed archive file '*logfile.0.gz'*. The logging utility then opens and writes to a new active log file. When the new active log file reaches the configured maximum size, '*logfile.0.gz'* is renamed '*logfile.1.gz'*, and the active log file is closed, compressed, and renamed '*logfile.0.gz'*. When the maximum number of archive files is reached and when the size of the active file reaches the configured maximum size, the contents of the oldest archived file are deleted so the current active file can be archived.

33 A 1Gb syslog file takes approximately 0.25Gb of storage when archived. Syslog files can acquire complete storage allocated to */var* filesystem, which is platform specific. However, when the filesystem reaches 92% storage capacity an event is raised to the administrator but the eventd process (being a privileged process) still can continue using the reserved storage blocks. This allows the syslog to continue storing events while the administrator frees the storage. If the administrator does not free the storage in time and the */var* filesystem storage becomes exhausted, a final entry is recorded in the log reporting "`No space left on device`" and logging is terminated. The appliance continues to operate in the event of exhaustion of audit log storage space.

6.2 Cryptographic Support (FCS)

6.2.1 FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1

34 FIPS-approved cryptographic functions implemented by the TOE are implemented in the following libraries:

- **Junos 24.2R2 – openSSL**
  (OpenSSL for Junos OS 24.2R2, based on version 1.1.1zb)
- **Junos 24.2R2 – libmd**
  (LibMD for Junos OS 24.2R2, created from same sources as OpenSSL version 1.1.1zb)
- **Junos 24.2R2 – kernel**
  (Kernel for Junos OS 24.2R2, based on FreeBSD-15 Stable release)

35 Each implementation of a cryptographic function by the TOE is CAVP validated. Only FIPS-approved cryptographic functions are used. Cryptographic algorithms and CAVP certificate mappings are described in Table 16.

Table 16: Cryptographic Algorithms and CAVP References

Library	SFR / Cryptographic Operation	Algorithm Capability	NIST Standard	Usage	CAVP
OpenSSL	FCS_COP.1 /DataEncryption (Encrypt, Decrypt)	AES-CBC (128, 256)	FIPS PUB 197 NIST SP 800-38A	SSH	A6931
		AES-CTR (128, 256)			
	FCS_COP.1 /Hash (Hashing)	SHA-256, SHA-384, SHA-512	FIPS PUB 180-4	SSH	
	FCS_COP.1 /KeyedHash (Keyed Hashing)	HMAC-SHA-256, HMAC-SHA-512	FIPS PUB 198-1	SSH	
	FCS_COP.1 /SigGen (SigGen, SigVer)	RSA (2048, 3072, 4096)	FIPS PUB 186-4 FIPS PUB 186-5	SSH Trusted Update	
		ECDSA (P-256, P-384, P-521	FIPS PUB 186-4		
	FCS_CKM.1 (KeyGen, KeyVer)	RSA (2048, 3072, 4096)	FIPS PUB 186-4 FIPS PUB 186-5	SSH	
		ECC (P-256, P-384, P-521)	FIPS PUB 186-4		
	FCS_CKM.2 (Key Establishment)	KAS-ECC-SSC	NIST SP 800-56A Revision 3	SSH	
LibMD	FCS_COP.1 /Hash (Hashing)	SHA-256,	FIPS PUB 180-4	Self-Tests	A6930
		SHA-512		Message Digest Generation in password hashing	

Library	SFR / Cryptographic Operation	Algorithm Capability	NIST Standard	Usage	CAVP
Kernel	FCS_COP.1 /Hash (Hashing)	SHA-1	FIPS PUB 180-4	Message Digest Generation in veriexec	A6929
		SHA-256, SHA-384		Self-Tests	
		SHA-512		RBG Functions	
	FCS_RBG_EXT.1 (DRBG)	HMAC_DRBG (HMAC-SHA-512)	NIST SP 800-90A	SSH (Random Bit Generation for key establishment)	

36 The FIPS approved algorithms are used when the FIPS mode is enabled. The relevant FIPS knobs are specified in [CCGUIDE]. The knob "`set system fips chassis level 1`" will enforce strict compliance to FIPS and enable restrictions on algorithms and keys sizes as required.

37 Asymmetric keys used by SSH are generated in accordance with FIPS PUB 186-4 Appendix B.3 for RSA Schemes and Appendix B.4 for ECC Schemes.

38 Junos OS handles zeroization for all CSP, plaintext secret and private cryptographic keys according to Table 17. (***FCS_CKM.4***).

Table 17: CSP Storage and Destruction

CSP	Description	Storage Method	Storage Location	Destruction Method
SSH Private Host Key	The first time SSH is configured, the key is generated. Used to identify the host.	Plaintext	File format on SDD (non-volatile)	Perform '*request system zeroize*' command.
	Loaded into memory to complete session establishment	Plaintext	Memory (Volatile)	*free()* is called by the TOE software at the session termination, or power cycle

CSP	Description	Storage Method	Storage Location	Destruction Method
SSH Session Key	Session keys used in SSH connections.	Plaintext	Memory (Volatile)	*free()* is called by the TOE software at the session termination, or power cycle
User/Administrator Password	Used to authenticate a user/administrator to the TOE.	Plaintext as entered	Processed in Memory (Volatile)	*free()* is called by the TOE software at the completion of authentication.
		Hashed (SHA-512)	Stored on disk (non-volatile)	Perform '*request system zeroize*' command.
RNG State	Internal state and seed key of RNG	Plaintext	Memory (Volatile)	Handled by kernel, overwritten with zero's at reboot.
SAK	MACsec Security Association Key	Encrypted	Memory (Volatile)	*free()* is called by the TOE software at the session termination, or power cycle
CAK	MACsec Connectivity Association Key	Obfuscated	Stored on disk (non-volatile)	Perform '*request system zeroize*' command.
KEK	MACsec key encryption key, derived from CAK.	Plaintext	Memory (Volatile)	*free()* is called by the TOE software at the session termination, or power cycle

39 The TOE implements the HMAC algorithms and values listed in Table 18.

Table 18: HMAC Values

	HMAC-SHA2-256	HMAC-SHA2-512
Key Length	256 bits	512 bits
Hash Function	SHA-256	SHA-512
Block Size	512 bits	1024 bits
Output MAC	256 bits	512 bits

40 Random number generation is implemented in accordance with NIST Special Publication 800-90 using HMAC_DRBG (SHA-512) implemented in the kernel library.

41 The platform-based entropy source generates at least 256 bits of entropy. The primary source of entropy is the RDSEED instruction of the Intel CPU within the TOE hardware. This entropy source is FIPS validated under ESV certificate #121. (***FCS_RBG_EXT.1***).

42 The TOE also includes the following additional secondary sources of entropy:

- Software Interrupts;
- Hardware Interrupts;
- Network Interrupts;
- TTY Interrupts.

43 In the evaluated configuration, SHA-512 is implemented in the LibMD library and used for password hashing by Junos' MGD daemon. The appliance must be operated with FIPS mode enabled.

6.2.2 FCS_NTP_EXT.1

44 The TOE supports NTPv4 using SHA-256 authentication. The TOE uses pre-shared symmetric keys for authentication and integrity of the NTP server when synchronizing the time.

6.2.3 FCS_SSH_EXT.1, FCS_SSHS_EXT.1

45 The TOE implements SSH in accordance with RFCs 4251, 4252, 4253, 4254, 4344, 5656, 6668, 8308, and 8332.

46 The TOE implements both public key and password-based authentication of administrative users and establishes the identity of a user by either verifying the SSH client public key in the authorized keys file, or by confirming the received username and password are valid credentials.

47 The TOE acts only as the server for SSH using the supported protocols and characteristics listed in Table 19.

Table 19: SSH Protocol Algorithms

Protocol	Key Exchange	Authentication	Data Encryption	Data Integrity
SSHv2	ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521	rsa-sha2-256 rsa-sha2-512 ecdsa-sha2-nistp256 ecdsa-sha2-nistp384 ecdsa-sha2-nistp521	aes128-ctr aes256-ctr aes128-cbc aes256-cbc	hmac-sha2-256 hmac-sha2-512

48 The HMAC algorithms in Table 19 use the values specified in Table 18.

49 The TOE drops all packets greater than 256KB in an SSH transport connection per RFC 4253.

50 The TOE implements SSH KDF per RFC 5656 Section 4.

51 The TOE performs a rekeying of SSH session keys if a one hour connection time has elapsed or after no more than 1 (one) gigabyte of data has been transmitted or received.

6.2.4 FCS_COP.1/CMAC, FCS_COP.1/MACSEC, FCS_MACSEC_EXT.1, FCS_MACSEC_EXT.2, FCS_MACSEC_EXT.3, FCS_MACSEC_EXT.4, FCS_MKA_EXT.1

52 FIPS-approved cryptographic functions implemented by the TOE for MACSEC operations are implemented in the following libraries:

- Junos 24.2R2 – Macsec
  (MACSEC Library for Junos OS 24.2R2)
- AES ECB 128bit & 256bit Encryption/Decryption Engine
  (MACsec FPGA(s))

53 The TOE implements MACSEC on a dedicated line card. The line card includes the hardware for handling the ports and dedicated firmware for implementing the MACSEC encryption.

54 MACsec cryptographic algorithms and CAVP certificate mappings are described in Table 20.

Table 20: MACsec Cryptographic Algorithms and CAVP References

Library	SFR / Cryptographic Operation	Cryptographic Operation	Algorithm Capability / NIST Standard	CAVP
MACSEC Library	FCS_COP.1 /MACSEC	Encryption/ Decryption	AES Key Wrap (Key Length: 128/256 bits) NIST SP 800- 38F	A6928

Library	SFR / Cryptographic Operation	Cryptographic Operation	Algorithm Capability / NIST Standard	CAVP
	FCS_COP.1 /CMAC	AES Keyed Hash Algorithm	AES-CMAC (Key Length: 128/256 bits MAC Length: 128 bits Block Size: 16 bytes) NIST SP 800-38B	
MACSEC Acceleration (MACSEC FPGA(s))	FCS_COP.1 /MACSEC	AES encryption and decryption (for BCM54192 and BCM54195 PHY)	AES-GCM (128, 256) FIPS PUB 197 NIST SP 800-38D	AES 4544
		AES encryption and decryption (for BCM82398 and BCM82399 PHY)		AES 4545
		AES encryption and decryption (for BCM82756 PHY)		AES 4550
		AES encryption and decryption (for BCM84894M PHY)		C996
		AES encryption and decryption (for BCM54998EM PHY)		C1869

55 MACsec is implemented in accordance with IEEE 802.1AE-2018 (***FCS_MACSEC_EXT.1***), supporting:

- AES 128/256 ciphersuite (without XPN)
- MACsec Key Agreement (MKA) protocol with Static-CAK mode using pre-shared key
- Connectivity-Association (CA) per physical port (IFD)
- 1 Tx-Secure Channel and 1 Rx- Secure Channel per CA
- 4 Secure Associations (SA) per SC

56 The Line Card can be programed to bypass certain ethertypes. In the evaluated configuration only Extended Authentication Protocol over LAN (EAPOL - PAE EtherType 88-8E), MACsec frames (EtherType 88-E5) and control frames

(EtherType is 88-08) are programmed to be bypassed. This means that only these Ethernet frames will be accepted by the TOE; all other frames will be rejected. Also, a filter in the packet forwarding engine (PFE) traps the packets to RE with ethertype 88-8E. (***FCS_MACSEC_EXT.1***)

57 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. (***FCS_MACSEC_EXT.1***)

58 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). Valid MKPDUs are decoded as described in section 11.11.4 of IEEE 802.1X-2010. Invalid MKPDUs are automatically discarded in the following circumstances which are consistent with FCS_MACSEC_EXT.1:

- the destination address of the MKPDU was an individual address;
- the MKPDU is less than 32 octets long;
- the MKPDU is not a multiple of 4 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; or
- the CAK Name is not recognized.

(***FCS_MKA_EXT.1***).

59 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 the MPDU. 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. (***FCS_MACSEC_EXT.2***)

60 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:

i) Offset 0 – Default; Encrypts the entire MPDU payload in the frame

ii) Offset 30 – IPv4 & TCP/UDP headers are unencrypted and rest of the payload is encrypted

iii) Offset 50 – IPv6 & TCP/UDP headers are unencrypted and rest of the payload is encrypted

61 The MKA is used to maintain MACsec Connectivity Association (CA). The TOE enforces MKA timeouts in accordance with IEEE 802.1X-2010 and 802.1Xbx-2014 as detailed in Table 21. Additionally, the TSF implements a timeout mechanism to ensure a maximum lifetime of two seconds for each MACsec frame. (***FCS_MKA_EXT.1***).

Table 21: MACsec MKA Timeout Values

Timer Use	Timeout (Parameter)	Timeout (Seconds)
Per participant periodic transmission, initialized on each transmission, transmission on expiry	MKA Hello Time	2.0
	MKA Bounded Hello Timeout	0.5
Per peer lifetime, initialized when adding to or refreshing the Potential Peers List or Live Peers List, expiry cause removal from the list.	MKA Life Time	6.0
Participant lifetime, initialized when participant created or following receipt of an MKPDU, expiry causes participant to be deleted.		
Delay after last distributing an SAK, before the Key Server will distribute a fresh SAK following a change in the Live Peer List while the Potential Peer List is still not empty.		

62 Each distributed SAK is protected by AES Key Wrap method with Key Encryption Key (KEK) as key input when transmitted (FCS_MACSEC_EXT.4). Each CAK is stored in an obfuscated form and there is no functions accessible for the administrator to read it (FPT_CAK_EXT.1). KEK is also derived from CAK that is PSK. Each participant that considers itself to be the current Key Server can distribute an SAK by encoding the following information in transmitted MKPDUs:

- The SAK protected by AES Key Wrap
- The Key Number (KN), 32 bits

63 A fresh SAK is not generated until the Key Server's Live Peer List contains at least one peer, and MKA Life Time has elapsed since the prior SAK was first distributed, or the Key Server's Potential Peer List is empty and PN number is exhausted.

64 SAK is generated using key derivation from CAK (AES-CMAC-128 or AES-CMAC-256) based on the cipher suite configured using the following transform function: (***FCS_MACSEC_EXT.3***):

SAK = KDF(Key, Label, KS-nonce | MI-value list | KN, SAKlength)

where

- Key = CAK.
- Labe l= "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.

65 The strength of the CAK and the size of the CAK's key space can be 128 bits or 256 bits.

6.3 Identification and Authentication (FIA)

6.3.1 FIA_AFL.1, FIA_PMG_EXT.1, FIA_UIA_EXT.1, FIA_UAU.7, FIA_PSK_EXT.1

66 The TOE accepts pre-shared CAKs for MACsec key agreement protocols as defined by IEEE 802.1X. The TSF accepts bit-based pre-shared keys entered as a string of up to 64 hexadecimal characters. (***FIA_PSK_EXT.1***).

67 The internal architecture supporting Authentication includes an active process, associated linked libraries and supporting configuration data. The Authentication process and library are:

- login()
- PAM Library module

68 Following TOE initialization, the login() process is listening for a connection at the local console. This 'login' process can be accessed through either direct connection to the local console or following successful establishment of a remote management connection over SSH, when a login prompt is displayed.

69 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).

70 The SSH daemon supports public key authentication by looking up a public key in an 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 client has a key available (***FIA_UIA_EXT.1***). The SSH daemon will ignore the authorized keys file if it or the directory '.ssh' or the user's home directory are not owned by the user or are writeable by anyone else.

71 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 (***FIA_UAU.7***). login() uses PAM Library calls for the actual verification of this password. The password is hashed and compared to the stored value, and success/failure is indicated to login(), (***FIA_UIA_EXT.1***). PAM is used to support authentication management, account management, session management and password management. Login primarily uses the session management and password management functionality offered by PAM.

72 *Retry-options* can be configured to specify the actions taken if an administrator fails to enter a valid username/password pair when authenticating via SSH. The *retry-options* are applied following the first failed login attempt for a given username (***FIA_AFL.1***).

73 The length of time (in seconds) following each failed attempt is specified by the *backoff-factor* which allows for a configurable range of 5 to 10, with the default being 5 (seconds).

74 The *backoff-threshold* is the threshold for the number of failed login attempts before the user experiences a delay in being able to enter a password again. The range is from 1 through 3, and the default value is 2 seconds.

75 The *tries-before-disconnect* sets the maximum number of times an administrator is allowed to enter a password to attempt to log in to the device through SSH before the connection is disconnected. The configurable range is between 2 and 10 attempts with a default value of 3.

76 The *lockout-period* sets the amount of time (in minutes) before an administrator can attempt to log in to the device after being locked out due to reaching the maximum number of failed login attempts. During this lockout period, an administrator is always able to login via the local console. The lockout-period is configurable to between 1 and 43,200 minutes.

77 The TOE requires users to enter correct identification and authentication data before any controlled access is granted. Prior to authentication, the TOE shall only allow displaying of an access banner, responding to an ICMP echo, and negotiation of a SSH session.

78 Passwords are case-sensitive, alphanumeric values. Passwords must achieve a minimum length of 10 characters, but is configurable to between 10 and 20 characters. All passwords must contain characters from at least two different character sets (upper, lower, numeric, punctuation). Any printable standard ASCII, extended ASCII and Unicode characters can be selected when choosing a password. (***FIA_PMG_EXT.1***).

6.4 Security Management (FMT)

6.4.1 FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services, FMT_MOF.1/Functions, FMT_MTD.1/CoreData, FMT_MTD.1/CryptoKeys, FMT_SMR.2, FMT_SMF.1, FMT_SMF.1/MACSEC

79 The TOE enforces binding between human users and subjects. The Security Administrator is responsible for provisioning user accounts, and only the Security Administrator can do so. (***FMT_SMR.2, FMT_MTD.1/CoreData***)

80 Users are configured under “system login user” and exported to the password database *‘/var/etc/master.passwd’*. A Junos user is an entry in the password database. Each entry in the password database has fields corresponding to the attributes of “system login user”, including username, (obfuscated) password and login class.

81 Accounts assigned to the Security Administrator role are used to manage Junos OS in accordance with NDcPP. User accounts in the TOE have the following attributes: user identity (user name), authentication data (password) and role (privilege). The Security Administrator is associated with the defined login class “security-admin”, which has the necessary permission set to permit the administrator to perform all tasks necessary to manage Junos OS in accordance with the requirements of NDcPP. (***FMT_SMR.2***).

82 The TOE allows user access either through the local console or remotely over SSH. Users are required to provide unique identification and authentication data before any access to the system is granted. (***FMT_SMR.2, FMT_SMF.1***).

83 The Security Administrator is capable of performing the following management functions:

- Ability to administer the TOE remotely via SSH;
- Ability to configure an access banner per FTA_TAB.1;

- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signatures prior to installation.
    - Initiate a manual update of TOE software per FMT_MOF.1/ManualUpdate;
    - Query currently executing version of TOE software (***FPT_TUD_EXT.1***)
    - Verify updates using digital signature (***FPT_TUD_EXT.1***)
- Ability to start and stop services;
    - All services that are configurable can be started and stopped via the CLI interface;
- Ability to configure local audit behaviour;
    - Handling of audit data, including setting limits of log file size (***FMT_MOF.1/Functions***)
- Ability to modify the behaviour of the transmission of audit data to an external IT entity;
    - Start/stop and modify the behaviour of the trusted communication channel to external syslog server (netconf over SSH) and the trusted path for remote Administrative sessions (SSH) (FMT_MOF.1/Functions, FMT_MOF.1/Services, FMT_SMF.1)
- Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;
- Ability to manage cryptographic keys (***FMT_MTD.1/CryptoKeys***);
    - Generate and delete SSH host keys.
- Ability to configure cryptographic functionality;
- Ability to configure thresholds for SSH rekeying;
- Ability to configure the time which is used for time-stamps;
- Ability to configure NTP;
- Ability to administer the TOE locally via the serial ports on the physical device;
- Ability to configure the local session inactivity time before session termination, including termination of session when local console cable is disconnected (FTA_SSL_EXT.1, FTA_SSL.3);
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to manage the trusted public keys database.

84 The Security Administrator is also capable of performing the following management functions related to MACsec functionality (***FMT_SMF.1/MACSEC***):

- Manage a PSK-based CAK and install it in the device;
- Manage the key server to create, delete, and activate MKA participants using the following CLI management command:

set security macsec interfaces <interface-name> connectivity-association <connectivity-association-name>

- Specify the lifetime of a CAK;
- Enable, disable, or delete a PSK-based CAK using the following CLI management commands:

  set security macsec connectivity-association <connectivity-association-name> pre-shared-key ckn <ckn>

  prompt security macsec connectivity-association <connectivity-association-name> pre-shared-key cak

6.5 Protection of the TSF (FPT)

6.5.1 FPT_SKP_EXT.1, FPT_APW_EXT.1, FPT_STM_EXT.1,

85 The CLI implemented by the TOE does not permit the viewing of cryptographic keys. There are no interfaces available that are designed specifically for the purpose of viewing keys and passwords in plaintext. All keys are protected through the enforcement of kernel-level file access rights which limit access to the contents of cryptographic key containers to processes with cryptographic rights or shell users with root permission. Security Administrators do not have root access rights to the kernel (***FPT_SKP_EXT.1***).

86 Locally stored authentication credentials are protected (***FPT_APW_EXT.1***):

- The password is hashed when stored using sha512.
- Authentication data for public key-based authentication methods are stored in a directory owned by the user (and typically with the same name as the user). This directory contains the files '.ssh/authorized_keys' and '.ssh/authorized_keys2' which are used for SSH public key authentication.

87 All events recorded by syslog are timestamped. The TOE supports synchronization with an NTP server or its own clock function within Junos OS that is maintained using the hardware Time Stamp Counter as the clock source. Both methods provide a source of date and time information that is free from outside interference for the following usages (***FPT_STM_EXT.1***):

- Audit log timestamps;
- Session timeouts and lockout enforcement;
- Cryptographic key regeneration intervals.

88 Security Administrators can configure NTP or manually configure the date and time during initial TOE configuration, and manually change the time during normal operation.

6.5.2 FPT_TST_EXT.1, FPT_FLS.1

89 The following self-tests are executed on power-on to verify the correct operation of the TOE software (***FPT_FLS.1, FPT_TST_EXT.1***):

90 **Power On Test** – At power-on, the TOE verifies that the boot-device responds, and performs a memory size check to confirm the amount of available memory. The TOE will then verify the integrity of all mounted signed packages, to assert that system files have not been tampered with. To test the integrity of the TOE software, the

fingerprints of the executables and other immutable files are regenerated and validated against the SHA1 fingerprints contained in the manifest file.

91 **Crypto Integrity Test** – checks integrity of major CSPs, such as SSH hostkeys.

92 **Authentication Error** – verifies that veriexec is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real.

93 **Known Answer Tests (KAT)** – verifies correct output from the following known answer tests:

- kernel—KAT for kernel cryptographic routines
- MACSec—KAT for MACsec cryptographic implementation
- libmd—KAT for libmd
- OpenSSL—KAT for OpenSSL cryptographic implementation

94 Juniper Networks devices run only binaries supplied by Juniper Networks. Within the package, each Junos OS software image includes fingerprints of the executables and other immutable files. Junos software will not execute any binary without a validating registered fingerprint. This feature protects the system against unauthorized software and activity that might compromise the integrity of the device. These self-tests ensure that only authorized executables are allowed to run thus ensuring the correct operation of the TOE.

95 In the event of a failure condition or a failure of any self-tests, the system will write the failure details to the system log file, enter the FIPS error state (panic), and then reboot. No command line input or traffic to any interface is processed during this time, and the device must successfully power cycle before attempting to return to normal operation. When the system restarts, the system boot process will not succeed without passing all applicable self-tests. This automatic recovery and self-test behavior is discussed in [CCGUIDE]. (***FPT_TST_EXT.1***)

6.5.3 FPT_TUD_EXT.1

96 Security Administrators are able to query the current version of the TOE software using the CLI command "*show version local*" (***FPT_TUD_EXT.1***) If a new version is available, they may initiate an update of the TOE software. Junos OS does not provide partial updates for the TOE. Updates are downloaded and applied manually. There is no automatic updating of the Junos OS. The installable software package containing the Junos OS has a digital signature that is checked when the Security Administrator attempts to install the package. (***FPT_TUD_EXT.1, FMT_SMF.1, FMT_MOF.1/ManualUpdate***,)

97 The Junos OS kernel maintains a set of fingerprints (SHA1 digests) for executable files and other files which should be immutable, as described in Section 6.5.2. The manifest file is signed in the production environment with the Juniper package signing key. The signature is verified by the TOE. ECDSA (P-256) with SHA-256 is used for digital signature package verification.

98 The fingerprint loader will only process a manifest for which it can verify the signature. Without a valid digital signature, an executable cannot be run. When the command is issued to install an update, the manifest file for the update is verified and stored, and each executable/immutable file is verified before being executed. If any of the fingerprints in an update are not correctly verified, the TOE uses the last known verified image. (***FPT_TUD_EXT.1***)

6.5.4 FPT_CAK_EXT.1, FPT_RPL.1

99 Each CAK is stored in an obfuscated form (9 format) which applies a string encoding mechanism that uses a Vigenère cipher. There is no function or interface available that is designed specifically for the purpose of viewing this information. (***FPT_CAK_EXT.1***).

100 The TOE is capable of detecting replayed data for MPDU and MKA frames and will discard the data and log the occurrence.

101 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. (***FPT_RPL.1***).

102 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). (***FPT_RPL.1***).

6.6 TOE Access (FTA)

6.6.1 FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4, FTA_TAB.1

103 The TOE allows Security Administrators to configure an access banner for local and remote SSH connections for display in the authentication prompt. The banner may display warnings against unauthorized access to the secure switch as well as any other information that the Security Administrator wishes to communicate. (***FTA_TAB.1***)

104 User sessions (local and remote) can be terminated by users (***FTA_SSL.4***). The administrative user can logout of existing CLI and remote SSH sessions by typing '*exit*' to exit the session and the TOE ensures that the current contents unreadable after the admin initiates the termination. No user activity can take place until the user re-identifies and authenticates.

105 Security Administrators may configure the TOE to terminate user sessions after a period of inactivity. (***FTA_SSL_EXT.1, FTA_SSL.3***)

106 For each user session the TOE maintains a count of clock cycles since the last activity. The count is reset each time there is activity related to the user session. When the counter reaches the number of clock cycles equating to the configured period of inactivity, the user session is locked out. The TOE also overwrites the display device and makes the current contents unreadable after the local interactive session is terminated due to inactivity, thus disabling any further interaction with the TOE.

6.7 Trusted Path/Channels (FTP)

6.7.1 FTP_ITC.1, FTP_TRP.1/Admin, FTP_ITC.1/MACSEC

107 The TOE implements a SSHv2 server. The TOE uses SSH for Trusted Channels between itself and a remote audit server and for Trusted Paths between itself and a remote management workstation. SSH connection protects the content of the communication from unauthorized disclosure or modification. (***FTP_ITC.1, FTP_TRP.1/Admin***)

108 Export of audit information to a secure, remote server is achieved by setting up an event trace monitor that sends event log messages by NETCONF over SSH to the remote audit server. The remote audit server initiates the connection. (***FTP_ITC.1***)

109 For remote administration, the remote administrator initiates communication with the TOE through a SSH tunnel created by a SSH session. Authentication of the peers is through public key cryptography. (***FTP_TRP.1/Admin***)

110 The TOE provides a secure channel between itself and MACsec peers that is logically distinct from other communication channels. All MACsec communications are protected against modification or disclosure of channel data and provides assured identification of peer end points. The TOE permits itself or another trusted IT entity to initiate communication via the trusted channel and the TOE will initiate communication with MACsec peers that require the use of MACsec. (***FTP_ITC.1/MACSEC***)

6.8 Flaw Remediation Procedures (ALC_FLR.3)

6.8.1 Flaw Reporting

111 Juniper implements a mature security flaw reporting and remediation process that ensures security flaws in the TOE are identified, examined, tracked, and corrected in a timely manner. The flaw remediation process also allows for TOE users and customers with an active support contract to register for support portal access where they can view information on discovered security flaws and official Juniper security advisories detailing corrective actions approved by Juniper.

112 Customers can report security flaws and vulnerabilities directly to the Juniper Networks Security Incident Response Team by any of the following methods:

- Email: sirt@juniper.net
- Phone: 888-314-JTAC FREE (North America) or +1-408-745-9500

113 Juniper takes a proactive approach to improving and securing its products while maintaining transparency. Juniper works in collaboration with groups inside and outside of Juniper to facilitate SIRT’s mission to address all security vulnerabilities.

6.8.2 Remediation Procedures

114 When a product reaches the End of Engineering (EOE) product lifecycle stage, six additional months will elapse until the product reaches End of Support (EOS) in the Juniper product life cycle. During this period, service tickets may still be generated for products, however SIRT will not track, confirm through testing, or produce advisories for vulnerabilities found after EOE in the product life cycle. For products within the active product life cycle, software releases may be categorized as a “code revision” which refers to maintenance or service releases (eg. 24.2R2 for maintenance or 24.2R1-S1 for service). “Supported Releases” include all software releases that have not yet reached end of engineering (EOE) support.

115 When a new security vulnerability is discovered or a suspected vulnerability report is received from an external source or via internal vulnerability monitoring teams, a Juniper Incident Manager (IM) will take ownership of the vulnerability in the GNATS defect tracking system, the system Juniper uses to track bugs and vulnerabilities. A development team will monitor security vulnerability reports for any third-party components that may be relevant while Juniper Product Security Engineers (PSE) work to replicate the bug or vulnerability on a representative sampling of impacted systems to confirm it is a vulnerability, be assessed for exploitability, and to subsequently document workarounds (when they exist), and the conditions that must

be present for the bug or vulnerability to be exploited. PSEs will then develop a fix that resolves the issue. Once a fix has been developed, it is applied to the test systems by SIRT PSEs and further regression testing is performed to confirm the fix addresses the bug or vulnerability on the impacted systems and does not otherwise negatively impact the systems or introduce further issues.

116 Once security vulnerabilities have been remediated and tested, the IM is responsible for the creation and publication of a Juniper Security Advisory (JSA). A resulting JSA includes detailed and robust information on the vulnerability and application of the fix. If a reported bug or suspected vulnerability is determined to not be a vulnerability or security-relevant, either through inability to reproduce or is found to be a configuration error, the issue is classified as 'Mistaken' and updated with an explanation that is also communicated to the reporting entity before being concluded.

117 While Juniper fixes all security vulnerabilities as quickly as possible, Juniper uses CVSS scoring to prioritize the security vulnerabilities to be resolved and the scope of the resulting fixes into the Juniper code bases:

- **CVSS score <3.0** – Low risk bugs or vulnerabilities where Juniper implements fixes in the current, mainline release. Rather than fixing every currently supported released, incorporating the fixes into the current, mainline release ensures the security shortcoming is repaired going forward.
- **CVSS scores >3.0 but <5.0.** Medium risk vulnerabilities where Juniper will fix the security bug in all supported releases.
- **CVSS scores >5.0.** High risk vulnerabilities. Juniper takes these security vulnerabilities very seriously. Security bugs with CVSS scores greater than or equal to 5.0 are considered "blockers" meaning that the fix for the shortcoming needs to be incorporated into the immediately following Juniper code revision. For such vulnerabilities, in addition to fixing the security flaw in all supported releases and incorporating it into the immediately following Juniper code revision, SIRT IM's will quarterly author and publish JSA's to describe these vulnerabilities. SIRT does so only after Juniper development engineers prepare and incorporate working fixes into all impacted code releases.

118 Juniper Security Advisories (JSA) are written descriptions that document a security vulnerability in a Juniper product or products. SIRT produces JSAs primarily for Juniper customers. The JSA provides information and actionable intelligence and guidance for customers about the security vulnerability.

119 With respect to JSAs and CVE IDs, there is either a 1:1 or 1:Many relationship. In general there is a 1:1 mapping of a single JSA to one unique CVE ID, however JSAs frequently contain more than one CVE ID. Juniper will publish JSAs in a bundle on a regular quarterly cadence to provide customers with regular and anticipated publications. If a security flaw is reported or discovered too close to the end of the Juniper SIRT quarterly JSA release cycle, it will be included in the following quarter's bundle. In exceptionally rare circumstances, the Juniper SIRT may publish an out-of-cycle Security Advisory, depending on the severity and active exploitation of a vulnerability. Juniper works to avoid publishing security advisories where a complete fix is not yet available, however for severe and actively exploited vulnerabilities, an advisory may be issued with only a workaround.

120 SIRT follows a set of processes and procedures to ensure consistency each quarter leading up to the publication of JSA's for higher severity vulnerabilities in Juniper Products as outlined in the table below

Timeframe	JSA Activity
T minus 90 days (i.e., T-90)	The newly appointed JSA bundle commander performs administrative tasks to ensure SIRT begins the new quarter with a blank slate.
T-45	By this point in the quarterly cycle, Juniper SIRT has visibility into and certainty about the set of problem reports (PRs) from the GNATS defect tracking system, with CVSS scores >= 5.0, that are likely to be in the upcoming JSA Bundle for the quarter. Aware of that, the IM bundle commander sends this list of PRs to Juniper Release Management (JRM) alerting them that SIRT intends to publish JSAs corresponding to each of these PRs at quarter end. The JRM team, separate from SIRT, then works with relevant stakeholders in Juniper to ensure those PRs are indeed fixed and that the fixes are incorporated into all impacted releases.
T-44 through T-35	Individual IMs prudently draft JSAs for each security vulnerability they shepherd through the process from discovery to remediation. SIRT IMs write JSAs with a keen awareness that they must balance the need to explain the vulnerability to defenders while being mindful not to inadvertently provide excess information to would-be attackers bent on exploiting and attacking weaknesses in Juniper products for nefarious purposes. In drafting JSAs, SIRT is meticulous about using clear, precise language. In addition to being conscientious and wary about providing too much information for attackers, another SIRT objective in drafting these advisories is to ensure a consistent look and feel from one SIRT-written JSA to the next, regardless of which SIRT IM team member drafted the security advisory.
T-32 through T-20	Juniper SIRT hosts daily, internal meetings during this period to carefully review each security advisory drafted by the IM team.
T-15	Juniper SIRT IMs send their draft JSAs to the engineer(s) that fixed the corresponding security bug, inquiring as to whether the IM properly characterized the vulnerability.
T-6	Juniper SIRT conducts an internal briefing for all interested internal stakeholders including Juniper's service, support, and account teams. SIRT announces to those in attendance the set of JSAs in the soon-to-be-published bundle, summarizing the ones most likely to be of interest to customers. Following the SIRT briefing, account teams have the remaining 6 days to prepare such that at T-0 the account team is in a position to discuss, with customers interested in this service, the details of relevant fixed vulnerabilities.
T-0	The SIRT IM, acting as bundle commander for the quarter, publishes the JSA Bundle (more details about publication fare in "JSA Bundle Publication").

121 The JSA bundle is published quarterly by an IM on the second Wednesday during the months of January, April, July, and October at 9am Pacific Time at https://advisory.juniper.net

7 Rationale

7.1 Conformance Claim Rationale

122 The following rationale is presented with regard to the PP/PP-Module conformance claims:

- **TOE type.** As identified in section 2.1, the TOE is a MACsec capable network device, compliant with CPP_ND_V3.0E, MOD_MACSEC_V1.0, and PKG_SSH_V1.0.
- **Security problem definition.** The threats, OSPs and assumptions are reproduced directly from CPP_ND_V3.0E, MOD_MACSEC_V1.0, and PKG_SSH_V1.0 if present.
- **Security objectives.** The security objectives are reproduced directly from CPP_ND_V3.0E, MOD_MACSEC_V1.0, and PKG_SSH_V1.0 if present.
- **Security requirements.** The security requirements are reproduced directly from CPP_ND_V3.0E, MOD_MACSEC_V1.0, and PKG_SSH_V1.0. No additional requirements have been specified.

7.2 Security Objectives Rationale

123 All security objectives are drawn directly from CPP_ND_V3.0E, MOD_MACSEC_V1.0, and PKG_SSH_V1.0 if present.

124 CPP_ND_V3.0E and PKG_SSH_V1.0 do not define any security objectives for the TOE, therefore those defined in MOD_MACSEC_V1.0 do not conflict with it.

125 MOD_MACSEC_V1.0 does not define any environmental objectives, but does note that OE.NO_THRU_TRAFFIC_PROTECTION from CPP_ND_V3.0E only applies to the Base-PP external interfaces. This is because the MACsec interface defined by this PP-Module does enforce through-traffic protection.

126 The following describes how the assumptions, threats, and organizational security policies map to the security objectives and is reproduced from MOD_MACSEC_V1.0 section 4.3.

Threat, Assumption, or OSP	Security Objectives	Rationale
T.DATA_ INTEGRITY	O.CRYPTOGRAPHIC_ FUNCTIONS_ MACSEC	The TOE mitigates the threat of data integrity violations by implementing cryptographic functionality that includes integrity protection.
	O.REPLAY_ DETECTION	The TOE mitigates the threat of data integrity violations by providing a mechanism to detect and discard replayed traffic for MPDUs.
T.NETWORK_ACCESS	O.PORT_FILTERING_ MACSEC	The TOE's port filtering capability reduces the threat of unauthorized access to devices in the TOE's

		operational environment by restricting the flow of network traffic entering through the TOE interfaces based on layer 2 frame characteristics and whether or not the traffic represents valid MACsec frames and MKPDUs.
T.UNTRUSTED_MACSEC_ COMMUNICATION_ CHANNELS	O.CRYPTOGRAPHIC_ FUNCTIONS_ MACSEC	The TOE mitigates the threat of unauthorized disclosure of information via untrusted thru traffic by providing MKA authentication functions to authorize endpoints.
T.UNAUTHORIZED_ ADMINISTRATOR_ ACCESS (from NDcPP)	O.AUTHORIZED_ ADMINISTRATION	The TOE further mitigates this threat originally defined in the Base-PP by defining additional management functions that require authorization and additional interfaces that can be used securely to execute management activities.
	O.AUTHENTICATION_ MACSEC	The TOE further mitigates this threat originally defined in the Base-PP by defining additional authentication requirements that establish connectivity between authenticated MACsec peers.
	O.TSF_INTEGRITY	The TOE further mitigates this threat originally defined in the Base-PP by implementing measures to fail securely if any self-test failures occur during startup, ensuring the device only operates when in a known state.
T.UNDETECTED_ACTIVITY (from NDcPP)	O.SYSTEM_ MONITORING_ MACSEC	The TOE further mitigates this threat originally defined in the Base-PP by implementing measures to generate audit records for security-relevant events that are specific to the functionality defined by this PP-Module.

Application Note: This table has been modified by TD0870.

7.3 Security Requirements Rationale

127 All security requirements are drawn directly from CPP_ND_V3.0E, MOD_MACSEC_V1.0, and PKG_SSH_V1.0.

128 See MOD_MACSEC_V1.0 section 6.1.4 for consistency rationale.

129 PKG_SSH_V1.0 does not provide a consistency rationale.

7.4 Security Problem Definition Rationale

130 The threats defined by MOD_MACSEC_V1.0 (See Table 7) supplement those defined in CPP_ND_V3.0E as follows:

PP-Module Threat, Assumption, OSP	Consistency Rationale
T.DATA_INTEGRITY	The threat of data integrity compromise at the layer 2 level is a specific threat that can be countered by MACsec technology.
T.NETWORK_ACCESS	The threat of a malicious entity accessing protected network resources without authorization is a specific example of the T.UNTRUSTED_COMMUNICATION_CHANNELS threat defined in the Base-PP.
T.UNTRUSTED_MACSEC_ COMMUNICATION_CHANNELS	The threat of disclosure of data in protected communications channels is the same as the T.UNTRUSTED_COMMUNICATION_CHANNELS threat in the NDcPP. This PP-Module expands on that by introducing additional logical interfaces (MACsec, SNMP) that this threat applies to.