Security Target Junos OS 22.4R2 for MX304, EX4100-48MP, EX4100-24MP, EX4100-24P, EX4100-24T, EX4100-48P, EX4100-48T

Juniper Networks

Version 1.1

February 22, 2024

Prepared for:
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net

Abstract

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), Junos OS 22.4R2 for MX304, EX4100-48MP, EX4100-24MP, EX4100-24P, EX4100-24T, EX4100-48P, EX4100-48T. This Security Target (ST) is conformant to the requirements of Collaborative Protection Profile for Network Devices v2.2E [NDcPP2.2E] and PP-Module for MACsec Ethernet Encryption Version 1.0 [MOD_MACSEC_V1.0].

References

[CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017

[CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.

[CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, CCMB-2017-04-004, Version 3.1, Revision 5, April 2017.

[CC_Add] CC and CEM Addenda, Exact Conformance, Selection-Based SFRs, Optional SFRs, CCDB-2017-05-xxx, Version 0.5, May 2017

[ECG-304] Junos OS Common Criteria Evaluated Configuration Guide for MX304 Device with JNP304-LMIC16 Line Card, Release 22.4R2, Published 2024-03-28

[ECG-4100] Junos OS Common Criteria Evaluated Configuration Guide for EX4100 Series Devices, Release 22.4R2, Published 2024-03-27

[NDcPP2.2E] Collaborative Protection Profile for Network Devices (NDcPP), Version 2.2E, 23-March-2020

[MOD_MACSEC_V1.0] PP-Module for MACsec Ethernet Encryption Version 1.0, dated 2023.03.02.

[SD] Supporting Document, Evaluation Activities for Network Device cPP, December-2019, version 2.2, CCDB-2019-12-004.

Table of Contents

1 Introduction

1. This section identifies the Security Target (ST), Target of Evaluation (TOE), Security Target organization, document conventions, and terminology. It also includes an overview of the TOE.

1.1 ST reference

ST Title	Security Target Junos OS 22.4R2 for MX304, EX4100-48MP, EX4100-24MP, EX4100-24P, EX4100-24T, EX4100-48P, EX4100-48T
ST Revision	1.1
ST Draft Date	February 22, 2024
Vendor	Juniper Networks, Inc.
cPP/EP Conformance	[NDcPP2.2E], [MOD_MACSEC_V1.0]

1.2 TOE Reference

TOE Title	Junos OS 22.4R2 for MX304, EX4100-48MP, EX4100-24MP, EX4100-24P, EX4100-24T, EX4100-48P, EX4100-48T
TOE Software	Junos OS 22.4R2
Security Guidance	1. Junos OS Common Criteria Evaluated Configuration Guide for MX304 Device with JNP304-LMIC16 Line Card, Release 22.4R2, Published 2024-03-28 2. Junos OS Common Criteria Evaluated Configuration Guide for EX4100 Series Devices, Release 22.4R2, Published 2024-03-27

1.3 About this document

2. This Security Target follows the following format:

Section	Title	Description
1	Introduction	Provides an overview of the TOE and defines the hardware and software that make up the TOE as well as the physical and logical boundaries of the TOE
2	Conformance Claims	Lists evaluation conformance to Common Criteria versions, Protection Profiles, or Packages where applicable
3	Security Problem Definition	Specifies the threats, assumptions and organizational security policies that affect the TOE
4	Security Objectives	Defines the security objectives for the TOE/operational environment and provides a rationale to demonstrate that the security objectives satisfy the threats
5	Security Functional Requirements	Contains the functional requirements for this TOE
6	Security Assurance Requirements	Contains the assurance requirements for this TOE
7	TOE Summary Specification	Identifies the IT security functions provided by the TOE and also identifies the assurance measures targeted to meet the assurance requirements
8	Acronyms	The acronyms used in the ST are explained

Table 1 Document Organization

1.4 Document Conventions

3. This document follows the same conventions as those applied in [NDcPP2.2E] in the completion of operations on Security Functional Requirements, namely:

- Unaltered SFRs are stated in the form used in [CC2] or their extended component definition;

- Refinement made in the ST: the refinement text is indicated with **bold text** and ~~strikethroughs~~;
- Selection completed in the ST: the selection values are indicated with underlined text

  e.g. "[*selection: disclosure, modification, loss of use*]" in [CC2] or an extended components definition might become "disclosure".
- Assignment completed in the ST: indicated with *italicized text*;
- Assignment completed within a selection in the ST: the completed assignment text is indicated with *italicized and underlined text*

  e.g. "[selection: *change_default, query, modify, delete, [assignment: other operations*]]" in [CC2] or an ECD might become "change_default, *select_tag*" (completion of both selection and assignment);
- Iteration: indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").

Where different notation is used in [MOD_MACSEC_V1.0], that notation shall be followed in those elements of the ST which are derived from [MOD_MACSEC_V1.0].

1.5 TOE Overview

4. The Target of Evaluation (TOE) is Juniper Networks, Inc. MX304, EX4100-48MP, EX4100-24MP, EX4100-24P, EX4100-24T, EX4100-48P, EX4100-48T 5G Universal Routing Platform executing the Junos OS 22.4R2 software.
5. The TOE is a complete virtual appliance consisting of all hardware, software and security guidance.
6. Each TOE is a secure network device that protects itself by offering only a minimal logical interface to the network and attached nodes. Junos OS 22.4R2 is a special purpose operating system that provides no general purpose computing capability. It implements both management and control functions as well as all IP routing.
7. The TOE allows definition and enforcement of information flow policies among subnetworks. Each information flow from one subnetwork to another passes through an instance of the TOE. The TOE makes a decision, based on the defined policies, whether the traffic is forwarded or dropped. Forwarding decisions are made on the basis of network addresses and protocols. The TOE also ensures that security-relevant activity is audited and provides the necessary functions to manage the security functions.
8. The TOE also implements Media Access Control Security (MACsec) encryption and decryption for a trusted channel at the Link Layer.

1.6 TOE Description

1.6.1 Overview

9. The TOE is an Ethernet-optimized edge router with 400-Gbps capacity. It implements both switching and carrier-class Ethernet routing. The TOE delivers an end-to-end infrastructure security solution for enterprises looking to move business-critical applications to public clouds. The TOE can be deployed in campus and branch access layer networks in the EVPN-VXLAN architectures.
10. The TOE is a complete routing system that delivers features, functionality, and secure services at scale in the 5G era. It shares common Junos software, features, and technology for compatibility across platforms.

11. The TOE is a physically self-contained appliance. It houses all software and hardware necessary to perform all routing functions. The architecture components of the TOE are:

    - Routing Engine (Control Board) – the Routing Engine (RE) runs the Junos OS 22.4R2 software and implements Layer 3 routing services and Layer 2 switching services. The RE also implements a network management interface for the configuration and operation of the TOE. The RE controls the flow of information through the TOE, including support for appliance interface control and control plane functions such as chassis component, system management and user access to the appliance.
    - The Packet Forwarding Engine (PFE) – implements all operations necessary for transit packet forwarding.
    - Power – power supply bays allow flexibility for provisioning and redundancy. The power supplies distribute the different output voltages produced by the power supplies to the TOE components depending on their voltage requirements.

12. The RE and the PFE function independently while constantly communicating through a high-speed internal link. This enables streamlined forwarding and routing control and the capability to run Internet-scale networks at high speeds.

13. The TOE can be administered using a Command Line Interface (CLI) through the Junos OS. The CLI can be accessed from a connected terminal console or over a network connection. Management over a network connection is secured using the SSH protocol. All management accesses require successful authentication.

14. The TOE implements EEE 802.1AE conformant Media Access Control Security (MACsec) for a Link Layer trusted channel between two instances of the TOE. The MACsec PHYs used by the variants of the TOE are given in the following:

TOE Variant	MACSEC PHY-1	MACSEC PHY-2
MX304	Juniper Trio 6	N/A
EX4100-48MP	BCM54998EM	BCM82756
EX4100-24MP	BCM84898M	BCM82756
EX4100-24P	BCM82756	N/A
EX4100-24T	BCM82756	N/A
EX4100-48P	BCM82756	N/A
EX4100-48T	BCM82756	N/A

1.6.2 Physical boundary

15. The TOE is the complete appliance consisting of the Junos OS 22.4R2 software running on the MX304, EX4100-48MP, EX4100-24MP, EX4100-24P, EX4100-24T, EX4100-48P, EX4100-48T chassis. The EX4100 Line variants of the TOE include an ARM-cortex A72 64-bit, single core processor. The MX304 variant includes an Intel Xeon D1735-TR CPU. The physical boundary of the TOE is the appliance chassis as illustrated in Figure 1. The TOE also includes a line card which implements MACsec. The line card used by the MX304 variant of the TOE is JNP304-LMIC16.

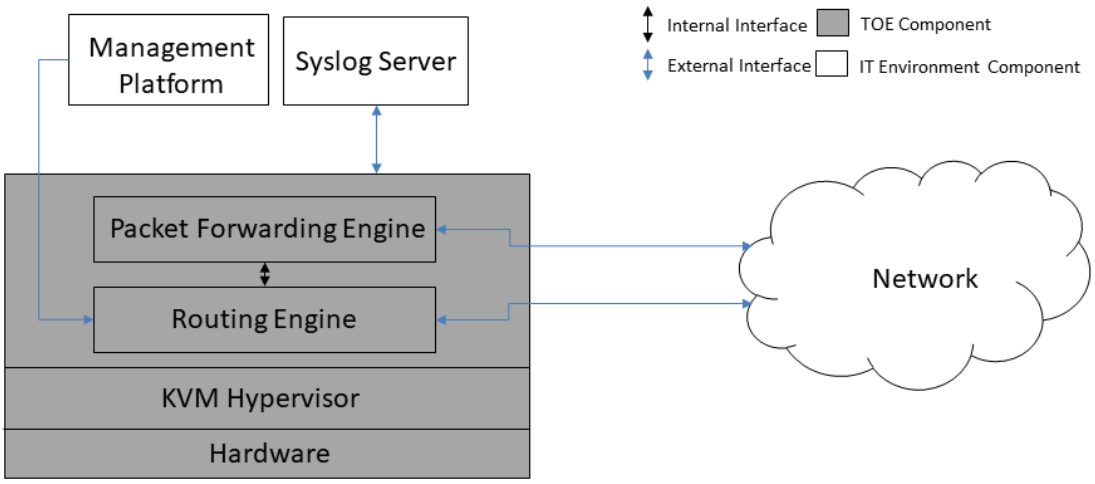

Figure 1 TOE Boundary

16. The interfaces the TOE are the network interfaces which control the traffic between the connected subnetworks and the management interface for administering the TOE.

17. The software images are the following:

    a. EX4100 variants of the TOE: junos-install-ex-arm-64-22.4R2.8.tgz.

    b. MX304 variant of the TOE: junos-vmhost-install-mx-x86-64-*22.4R2.8*.tgz.

18. The software version can be viewed by an administrator by the `show version` command executed on the CLI of the TOE.

19. The guidance documents included in the physical scope as part of the TOE are [ECG-304] and [ECG-4100].

1.6.3 Logical Scope of the TOE

20. The logical boundary of the TOE includes the following security functionalities:

Security Functionality	Description
Security Audit	The TOE generates an audit record for each auditable event. The audit records are stored in syslog files on the TOE and can be sent to an external audit server via Netconf over SSH. Auditable events include start-up and shutdown of the audit functions, authentication events, and all events listed in Table 9. Audit records include the date and time of the event, event category, event type, username, and the outcome of the event (success or failure). Local syslog storage limits are configurable and are monitored. If the storage limit is reached the oldest logs will be overwritten.
Cryptographic Support	The TOE implements an SSH server for administrators to establish secure sessions between the network management station and the TOE, and to connect to external audit servers. Each remote host must be successfully authenticated prior to the TOE allowing any communication with it. The TOE includes cryptographic modules that implement the underlying cryptographic services, including key management and protection of stored keys, cryptographic algorithms, random bit generation and administration of the cryptographic

	functions. SSH implemented with the cryptographic modules enforce authenticity, confidentiality and integrity of all communication and administrative accesses to the TOE.
Identification and Authentication	The TOE implements Role Based Access Control. Each human user must be successfully authenticated and assigned to the role Security Administrator by the TOE prior to the TOE granting them access to the CLI for the management of the TOE. Human users are authenticated with a password while the remote hosts are authenticated with public key cryptography. Authentication data entered and stored on the TOE is protected. The TOE can be configured to terminate interactive user sessions and to present an access banner with warning messages prior to authentication.
Security Management	The TOE implements a Security Administrator role. Users successfully authenticated and assigned to the role are granted the right to: • configuration and maintenance of cryptographic functions used for the establishment of secure connections to and from the TOE; • review all audit data; • initiation of trusted updates; and • all other administrative tasks. The TOE is managed through a Command Line Interface (CLI) which is accessible through local (serial) console or remotely over SSH.
Protection of the TSF	The TOE protects all passwords, pre-shared keys, symmetric keys and private keys from unauthorized disclosure. Passwords are stored using SHA-1, SHA-256 or SHA-512. The TOE executes a suite of self-tests during the initial start-up to ensure the correct operation of critical security functions. All software updates may be verified for authenticity. The TOE also implements an internal clock to maintain the date and time and to issue time stamps for other security functions. If the MACsec functionality fails for any reason, the TOE will always restore a secure state.
TOE Access	The TOE displays an access banner in each user authentication exchange. The banner messaging is customizable to allow the Security Administrator to inform the users of the conditions of access and sanctions of attempted unauthorized access. The TOE maintains an inactivity timer for each session and will terminate each interactive session after a period of inactivity. The CLI implements a command `exit` which allows each user to terminate their session.
Trusted Path/Trusted Channel	The TOE implements a SSHv2 server. SSH is used for secure communication between the TOE and a remote Syslog server and between the TOE and a network management station. The TOE also implements EEE 802.1AE conformant MACsec for link layer trusted channels.

Table 2 Logical Scope of TOE

1.6.4 Non-TOE hardware/software/firmware

21. The TOE requires the following elements in the network environment:

- Syslog server supporting SSHv2 connections for sending audit logs;
- A Management station with a SSHv2 client for remote administration; and
- Serial connection client for local administration.

1.6.5 Summary of out scope items

- Use of telnet, since it violates the Trusted Path requirement set (see Section 5.7.2)
- Use of FTP, since it violates the Trusted Path requirement set (see Section 5.7.2)
- Use of SNMP, since it violates the Trusted Path requirement set (see Section 5.7.2)
- Use of SSL, including management via J-Web, JUNOScript and JUNOScope, since it violates the Trusted Path requirement set (see Section 5.7.2)
- Use of CLI account super-user and linux root account.

2 Conformance Claim

2.1 CC Conformance Claim

22. The TOE and ST are compliant with the Common Criteria (CC) Version 3.1, Revision 5, dated: April 2017. This ST and the TOE are conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017: Part 3 conformant

2.2 PP Conformance claim

23. This TOE is conformant to:

- Collaborative Protection Profile for Network Devices (NDcPP), Version 2.2e 23-March-2020
- PP-Module for MACsec Ethernet Encryption Version 1.0, dated 2023.03.02

24. The above are applied in accordance with the following:

- PP-Configuration for Network Devices and MACsec Ethernet Encryption, 2023-03-29 (CFG_NDcPP-MACsec_V1.00

2.3 Conformance Rationale

25. This Security Target claims exact conformance to Version 2.2E of the Collaborative Protection Profile for Network Devices. The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile. Only operations allowed in the Protection Profile are performed on the Security Requirements.

26. As per the Conformance Claim of [MOD_MACSEC_V1.0], the PP-Module inherits exact conformance as required from the specified Base-PP and as defined in the CC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017). Consequently, there are no contradictions between the Base-PP and the PP-Module.

2.4 Technical Decisions

27. The NIAP Technical Decisions (TD) applicable to the TOE are listed in Table 3.

Technical Decisions	Applicable	Exclusion Rationale (if applicable)
TD0800 – Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	No	The TOE does not claim IPsec or IKE.
TD0792 – NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes	
TD0790 – NIT Technical Decision: Clarification Required for testing IPv6	No	The TOE does not implement IPv6, TLS or DTLS.
TD0748 – Correction to FMT_SMF.1/MACSEC Test 21	Yes	
TD0746 – Correction to FPT_RPL.1 Test 25	Yes	
TD0738 – NIT Technical Decision for Link to Allowed-With List	Yes	

TD0728 – Corrections to MACSec PP-Module SD	Yes	
0670 – NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	No	The TOE does not claim TLS.
0639 – NIT Technical Decision for Clarification for NTP MAC Keys	No	The TOE does not claim NTP.
0638 – NIT Technical Decision for Key Pair Generation for Authentication	Yes	
0636 – NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No	The TOE does not claim SSH Client.
0635 – NIT Technical Decision for TLS Server and Key Agreement Parameters	No	The TOE does not claim TLS.
0634 – NIT Technical Decision for Clarification required for testing IPv6	No	Superseded by TD 0790.
0633 – NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	No	The TOE does not claim IPv6.
0632 – NIT Technical Decision for Consistency with Time Data for vNDs	Yes[1]	
0631 – NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes	
0592 – NIT Technical Decision for Local Storage of Audit Records	Yes	
0591 – NIT Technical Decision for Virtual TOEs and hypervisors	Yes	
0581 – NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
0580 – NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
0572 – NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
0571 – NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
0570 – NiT Technical Decision for Clarification about FIA_AFL.1	Yes	
0569 – NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	The TOE does not claim DTLS.
0564 – NiT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
0563 – NiT Technical Decision for Clarification of audit date information	Yes	
0556 – NIT Technical Decision for RFC 5077 question	No	The TOE does not claim TLS.
0555 – NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	The TOE does not claim TLS.
0547 – NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	

[1] The TOE is not a virtual Network Device (vND) but the TD requires modification of FPT_STM_EXT.1.2 which the TOE claims. The selection implemented in this ST does not claim the vND specific items.

0546 – NIT Technical Decision for DTLS - clarification of Application Note 63	No	The TOE does not claim DTLS.
0537 – NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	No	The TOE does not claim TLS.
0536 – NIT Technical Decision for Update Verification Inconsistency	Yes	
0528 – NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No	The TOE does not claim NTP.
0527 – Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	No	The TOE does not claim X.509 certificate based authentication.

Table 3 Technical Decisions

3 Security Problem Definition

28. The security problem definition has been taken from [NDcPP2.2E] and [MOD_MACSEC_V1.0], and is reproduced here.

3.1 Threats

29. Threats applicable to the TOE are given in Table 4.

Threat	Threat Definition
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 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 include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_FUNCTIONALITY _FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.
T.DATA_INTEGRITY	**An attacker may modify data transmitted over the layer 2 link in a way that is not detected by the recipient.** Devices on a network may be exposed to attacks that attempt to corrupt or modify data in transit without authorization. If malicious devices are able to modify and replay data that is transmitted over a layer 2 link, then the data contained within the communications may be susceptible to a loss of integrity.
T.NETWORK_ACCESS	**An attacker may send traffic through the TOE that enables them to access devices in the TOE's operational environment without authorization.** **A MACsec device may 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 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.

Table 4 Threats

3.2 Assumptions

30. The assumptions applicable to the TOE are given in Table 5.

Assumption	Assumption Definition
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 and/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 will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will 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.
A.LIMITED _FUNCTIONALITY[2]	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 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).
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 being 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

[2] In accordance with TD0591

	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.

Table 5 Assumptions

3.3 Organizational Security Policies

31. The OSPs applicable to the TOE are given in Table 6.

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

Table 6 Organizational Security Policies

4 Security Objectives

32. The security objectives have been taken from [NDcPP2.2E] and [MOD_MACSEC_V1.0], and are reproduced here.

4.1 Security Objectives for the TOE

33. The security objectives for the TOE derived from the Base-PP are trivially determined through the inverse of the statement of threats presented in Sect. 4.1 of [NDcPP2.2E] and are not explicitly stated.

34. The Security Objectives for the TOE drawn from [MOD_MACSEC_V1.0] are given in Table 7.

Security Objective for the TOE	Security Objective Definition
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

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 7 Security Objectives for the TOE

4.2 Security Objectives for the Operational Environment

35. Security objectives for the Operational Environment are given in Table 8.

Environment Security Objective	Security Objective Definition
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.
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	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Table 8 Security Objectives for Operational Environment

4.3 Security Objectives rationale

36. Security objectives for the TOE and for the operational environment are taken from [NDcPP2.2E] and [MOD_MACSEC_V1.0] and reproduced exactly. Therefore, security objectives rationale is identical to that given in Sect. 4 of [NDcPP2.2E] and Sect. 4.3 of [MOD_MACSEC_V1.0]. it is not reproduced here.

5 Security Functional Requirements

37. All security functional requirements are taken from the [NDcPP2.2E] and [MOD_MACSEC_V1.]. They are presented in accordance with the conventions described in Sect. 1.4. Extended component definitions are given in Annex C of [NDcPP2.2E] and Appendix C of [MOD_MACSEC_V1.0]. They are not repeated here. Security Requirements Rationales are also identical to those given in [NDcPP2.2E] and [MOD_MACSEC_V1.] and are not repeated.

5.1 Security Audit (FAU)

5.1.1 Security Audit Data generation (FAU_GEN)

5.1.1.1 FAU_GEN.1 Audit data generation

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 user account shall be logged if individual user 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 user account shall be logged).*
- *[no other actions];*

d) Specifically defined auditable events listed in Table 9.

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, 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 ST*, information specified in column three of Table 9*.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	None	None
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_RBG_EXT.1	None	None
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 mechanism.	Origin of the attempt (e.g., IP address)
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address)
FIA_UAU.7	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None
FMT_MTD.1/CoreData	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 interactive session by the session locking mechanism.	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
FTP_ITC.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failure of the trusted path functions.	None.
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FMT_MOF.1/Functions	None.	None.

FMT_MOF.1/Services	None.	None
FMT_MTD.1/CryptoKeys	None.	None.

Table 9 Security Functional Requirements and Auditable Events

5.1.1.2 FAU_GEN.1 Audit data generation

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 shut-down 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 Auditable Events Table (***Table 10***)]***.

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

Table 10 Auditable Events

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 10)].

5.1.1.3 FAU_GEN.2 User identity association

FAU_GEN.2 User identity association

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

5.1.2 Security audit event storage (Extended – FAU_STG_EXT)

5.1.2.1 FAU_ STG_EXT.1 Protected Audit Event Storage

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 [

- *TOE shall consist of a single standalone component that stores audit data locally].*

FAU_STG_EXT.1.3 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.

5.1.2.2 FAU_STG.1 Protected audit trail storage (Optional)

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.

5.2 Cryptographic Support (FCS)

5.2.1 Cryptographic Key Management (FCS_CKM)

5.2.1.1 FCS_CKM.1 Cryptographic Key Generation (Refinement)

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-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*
- *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;*
- *FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526].*

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

5.2.1.2 FCS_CKM.2 Cryptographic Key Establishment (Refinement)

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1[3] 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 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";*
- *FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [groups listed in RFC 3526].;*

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

5.2.1.3 FCS_CKM.4 Cryptographic Key Destruction

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];*

[3] In accordance with TD0580, TD0581

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

that meets the following: *No Standard*.

5.2.2 Cryptographic Operation (FCS_COP)

5.2.2.1 FCS_COP.1 Cryptographic Operation

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 and cryptographic key sizes (modulus) [2048 bits, 3072 bits, 4096 bits],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [P-256, P-384, P-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 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: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4*

].

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-1, HMAC-SHA-256, HMAC-SHA-512*] and cryptographic key sizes [*160, 256 and 512 bits*] **and message digest sizes [*160, 256, 512*] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

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*].

Application Note: AES-CMAC is a keyed hash function that is used as part of the key derivation function (KDF) that is used for key generation.

FCS_COP.1/MACSEC Cryptographic Operation (MACsec AES Data Encryption and Decryption)

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

5.2.2.2 FCS_MACSEC_EXT.1 MACsec

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: Depending on the Carrier Ethernet service provider a TOE might need basic VLAN tag handling abilities such as a simple add or discard to be suitable for Use Case 2.

5.2.2.3 FCS_MACSEC_EXT.2 MACsec Integrity and Confidentiality

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.

Application Note: The length of the ICV is dependent on the ciphersuite used but will not be less than 8 octets or more than 16 octets at the end of the MPDU. The ICV protects the destination and source MAC address parameters, as well as all the fields of the MPDU.

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

5.2.2.4 FCS_MACSEC_EXT.3 MACsec Randomness

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-2010*] 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: FCS_RBG_EXT.1 is defined in the Base-PP so a conformant MACsec TOE will include this dependency.

5.2.2.5 FCS_MACSEC_EXT.4 MACsec Key Usage

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*].

Application Note: The definition of the peer's CAK as defined by IEEE 802.1X- 2010 is synonymous with the peer authentication performed here. If "EAP-TLS with DevIDs" is selected, the FCS_DEVID_EXT.1 and FCS_EAPTLS_EXT.1 SFRs must be claimed.

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

Application Note: This requirement applies to the SAKs that are generated by the TOE. They must be wrapped by the AES Key Wrap method specified in NIST SP 800-38F.

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-2010, Section 9.8.1).

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

5.2.2.6 FCS_MKA_EXT.1 MACsec Key Agreement

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-2010 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).

Application Note: The ICV has length 128 bits and is computed according to Section 9.4.1 of IEEE 802.1X-2010. The ICV protects the destination and source MAC address parameters, as well as all the fields of the MAC Service Data Unit of the MKPDU including the allocated EtherType, and up to but not including, the generated ICV.

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 Bounded Hello Timeout limit of 0.5 seconds.

Application Note: The key server may also distribute a group CAK established by pairwise CAKs.

FCS_MKA_EXT.1.5 The key server shall refresh a SAK when it expires. The key server shall distribute a SAK by [

- *distributed by pre-shared key (PSK)*

].

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-2010 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-2010 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-2010 Section 9.4.1 shall be decoded as specified in IEEE 802.1X-2010 Section 11.11.4.

5.2.3 FCS_RBG_EXT.1 Random Bit Generation

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 (any)*].

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

5.2.4 Cryptographic Protocols (Extended – FCS_SSHS_EXT SSH Protocol)

5.2.4.1 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) [*4251, 4252, 4253, 4254, 4344, 5656, 6668*].

FCS_SSHS_EXT.1.2[4] The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [*password-based*].

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

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [*aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr*].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [*ssh-rsa, rsa-sha2-256, rsa-sha2-512,ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [*hmac-sha1, hmac-sha2-256, hmac-sha2-512*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

[4] In accordance with TD0631

FCS_SSHS_EXT.1.7 The TSF shall ensure that [*diffie-hellman-group14-sha1, ecdh-sha2-nistp256*] and [*ecdh-sha2-nistp384, ecdh-sha2-nistp521*] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

5.3 Identification and Authentication (FIA)

5.3.1 Authentication Failure Management (FIA_AFL)

5.3.1.1 FIA_AFL.1 Authentication Failure Management (Refinement)

FIA_AFL.1 Authentication Failure Management

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

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall [p*revent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed*].

5.3.1.2 FIA_PSK_EXT.1 Pre-Shared Key Composition

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-2010, [*no other protocols*].

Application Note: If other protocols can use PSKs, they should be listed in the assignment as well; otherwise “no other protocols” should be chosen.

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

Application Note: The ST author specifies whether the TSF merely accepts bit-based PSKs or if it is also capable of generating them. If it generates them, the requirement specifies that they must be generated using the RBG provided by the TOE.

5.3.2 Password Management (Extended – FIA_PMG_EXT)

5.3.2.1 FIA_PMG_EXT.1 Password Management

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 standard ASCII, extended ASCII and Unicode characters*]];
b) Minimum password length shall be configurable to between [*10*] and [*20*] characters.

5.3.3 User Identification and Authentication (Extended – FIA_UIA_EXT)

5.3.3.1 FIA_UIA_EXT.1 User Identification and Authentication

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.

5.3.4 User authentication (FIA_UAU) (Extended – FIA_UAU_EXT)

5.3.4.1 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local [*password-based*] authentication mechanism to perform local administrative user authentication.

5.3.4.2 FIA_UAU.7 Protected Authentication Feedback

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**.

5.4 Security Management (FMT)

5.4.1 Management of functions in TSF (FMT_MOF)

5.4.1.1 FMT_MOF.1/ManualUpdate Management of security functions behaviour

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*.

5.4.1.2 FMT_MOF.1/Services Management of security functions behaviour

FMT_MOF.1/Services Management of security functions behaviour

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

5.4.1.3 FMT_MOF.1/Functions Management of security functions behaviour

FMT_MOF.1/Functions Management of security functions behaviour

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

5.4.2 Management of TSF Data (FMT_MTD)

5.4.2.1 FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1/CoreData Management of TSF Data

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

5.4.2.2 FMT_MTD.1/CryptoKeys Management of TSF data

FMT_MTD.1/CryptoKeys Management of TSF data

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

5.4.3 Specification of Management Functions (FMT_SMF)

5.4.3.1 FMT_SMF.1 Specification of Management Functions

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 locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*

[

  - *Ability to start and stop services;*
  - *Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);*
  - *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
  - *Ability to manage the cryptographic keys;*
  - *Ability to configure the cryptographic functionality;*
  - *Ability to configure thresholds for SSH rekeying;*
  - *Ability to re-enable an Administrator account;*
  - *Ability to set the time which is used for time-stamps;*
  - *Ability to manage the trusted public keys database[5].*

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

*[**selection:***

- *No other MACsec management functions*

]].

[5] In accordance with TD0631

Application Note: IEEE 802.1X-2010 specifies Management Information Base (MIB) objects for management functionality but configuration of management functions via other approved methods is acceptable. The ST author should select either the MIB object or provide the function used to achieve this management functionality.
If a selection containing "group CAK" is chosen in FCS_MKA_EXT.1.5, then "Cause key server to generate a new group CAK…" must be selected.

5.4.4 Security management roles (FMT_SMR)

5.4.4.1 FMT_SMR.2 Restrictions on security roles

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 locally;*
- *The Security Administrator role shall be able to administer the TOE remotely*

are satisfied.

5.5 Protection of the TSF (FPT)

5.5.1 Protection of TSF Data (Extended – FPT_SKP_EXT)

5.5.1.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

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

5.5.2 Protection of Administrator Passwords (Extended – FPT_APW_EXT)

5.5.2.1 FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1 Protection of Administrator Passwords

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

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

5.5.3 TSF testing (Extended – FPT_TST_EXT)

5.5.3.1 FPT_TST_EXT.1 TSF Testing (Extended)

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 demonstrate the correct operation of the TSF: [*Power on test, File integrity test, Crypto integrity test, Authentication test, Algorithm known answer tests*].

5.5.4 Trusted Update (FPT_TUD_EXT)

5.5.4.1 FPT_TUD_EXT.1 Trusted Update

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.

5.5.5 Time stamps (Extended – FPT_STM_EXT))

5.5.5.1 FPT_STM_EXT.1 Reliable Time Stamps

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[6] The TSF shall [*allow the Security Administrator to set the time*].

5.5.5.2 FPT_DDP_EXT.1 Data Delay Protection

FPT_DDP_EXT.1 Data Delay Protection

FPT_DDP_EXT.1.1 The TSF shall enable data delay protection for MKA that ensures data frames protected by MACsec are not delayed by more than two seconds.

5.5.5.3 FPT_CAK_EXT.1 Protection of CAK Data

FPT_CAK_EXT.1 Protection of CAK Data

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

Application Note: The intent is for the TOE to protect CAK data from unauthorized disclosure. This data should only be accessed for the purposes of its assigned security functionality and there is no need for it to be displayed or accessed at any other time. This requirement does not prevent the device from providing indication that these exist, are in use, or are still valid. It does, however, restrict the reading of the values outright.

5.5.5.4 FPT_FLS.1 Failure with Preservation of Secure State

FPT_FLS.1 Failure with Preservation of Secure State

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

Application Note: The intent of this requirement is to express the fail secure capabilities that the TOE possesses. This means that the TOE must be able to attain a secure/safe state (shutdown) when any of the identified failures occur. For a TOE with redundant failover capability (that continues to operate if poweron self-test (POST) passes on the redundant component), in the event of a POST failure on a redundant component, the specific component that received the POST failure will be

[6] In accordance with TD0632

shut down. For conformance with other PP-Modules it might be a requirement for the fail-secure state to be "shut down."

5.5.5.5 FPT_RPL.1 Replay Detection

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.

Application Note: As per IEEE 802.1AE-2018, replay is detected by examining the PN value that is embedded in the SecTag that is at the header of the MPDU. The PN is encoded in octets 5 through 8 of the SecTag to support replay protection.

5.6 TOE Access (FTA)

5.6.1 TSF-initiated Session Locking (Extended – FTA_SSL_EXT)

5.6.1.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1 TSF-initiated Session Locking

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

- *terminate the session*]

after a Security Administrator-specified time period of inactivity.

5.6.2 Session locking and termination (FTA_SSL)

5.6.2.1 FTA_SSL.3 TSF-initiated Termination (Refinement)

FTA_SSL.3 TSF-initiated Termination

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

5.6.2.2 FTA_SSL.4 User-initiated Termination (Refinement)

FTA_SSL.4 User-initiated Termination

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

5.6.3 TOE access banners (FTA_TAB)

5.6.3.1 FTA_TAB.1 Default TOE Access Banners (Refinement)

FTA_TAB.1 Default TOE Access Banners (Refinement)

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

5.7 Trusted path/channels (FTP)

5.7.1 Trusted Channel (FTP_ITC)

5.7.1.1 FTP_ITC.1 Inter-TSF trusted channel (Refinement)

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 **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 **disclosure and detection of modification of the channel data**.

FTP_ITC.1.2 The TSF shall permit **the TSF or 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 communication*].

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*].

5.7.2 Trusted Path (FTP_TRP)

5.7.2.1 FTP_TRP.1/Admin Trusted Path (Refinement)

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** 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** 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*.

6 Security Assurance Requirements

38. The security assurance requirements are from Sect. 7 of [NDcPP2.2E] and listed in Table 11. The developer of the TOE implements corresponding assurance measures in the development of the TOE to address each assurance requirement.

Assurance Class	Assurance Component
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)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

Table 11 Security Assurance Requirements

7 TOE Summary Specification

7.1 Protected communications

39. Local console access is gained by connecting an RJ-45 cable between the console port on the TOE and a workstation with a serial connection client.

7.1.1 Algorithms and zeroization

40. 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. Other FIPS-approved cryptographic functions implemented by the TOE are implemented in the following libraries:

    - OpenSSL for Junos OS 22.4R2 (based on version 1.0.2p)
    - LibMD for Junos OS 22.4R2 (created from same sources as OpenSSL version 1.0.2p)
    - Kernel for Junos OS 22.4R2 (based on FreeBSD-11 Stable release)

41. Random number generation is implemented in accordance with NIST Special Publication 800-90 using HMAC_DRBG implemented in the OpenSSL library and kernel library. Each entropy source is software-based and generates at least 256 bits of entropy (**FCS_RBG_EXT.1**). Additionally, SHA-256 and SHA-512 are implemented in the LibMD library and used for password hashing by Junos' MGD daemon. **The appliance is to be operated with FIPS mode enabled.**

42. Each implementation of a cryptographic function by the TOE is CAVP validated. Only FIPS-approved cryptographic functions are used. CAVP certificate references are given in Table 12.

Library	NIST Standard	Algorithm, Mode, Keysize, Function, Hashing, Usage	Cryptographic Operation	SFR(s) supported	CAVP Reference
OpenSSL	FIPS 197, SP 800-38A	AES-CBC (128, 256)	Encrypt, Decrypt in SSH	FCS_COP.1/DataEncryption FCS_SSHS_EXT.1	A4301
	FIPS 197, SP800-38A	AES-CTR (128, 256)	Encrypt, Decrypt in SSH	FCS_COP.1/DataEncryption FCS_SSHS_EXT.1	A4301
	FIPS 180-4	SHA1, SHA-256, SHA-384, SHA-512 (byte Oriented)	Message Digest Generation in SSH	FCS_CKM.2 FCS_COP.1/Hash FCS_SSHS_EXT.1	A4301
	FIPS 198-1	HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-512 (byte Oriented)	Message Authentication in SSH and DRBG primitive for OpenSSL DRBG	FCS_COP.1/KeyedHash FCS_SSHS_EXT.1	A4301
	FIPS 186-4	ECDSA (P-256 w/ SHA-256) ECDSA (P-384 w/ SHA-384) ECDSA (P-521 w/ SHA-521)	SigGen, SigVer, KeyGen for ECDSA in SSH	FCS_CKM.1 FCS_COP.1/SigGen FCS_SSHS_EXT.1 FPT_TUD_EXT.1	A4301
	SP800-56A	CVL/KAS ECC Key Agreement EC (P-256, SHA-256), ED (P-384, SHA-384), EE (P-521, SHA-512	Public key Validation, Key Pair Generation, Initiator and Responder for SSH ECDH	FCS_CKM.1 FCS_CKM.2 FCS_COP.1/SigGen FCS_SSHS_EXT.1	A4301
	FIPS 186-4	RSA PKCS1_V1_5[7] (n=2048 (SHA-256), n=3072 (SHA-256), n=4096 (SHA-256))	KeyGen, SigGen, SigVer in SSH	FCS_CKM.1 FCS_COP.1/SigGen FCS_SSHS_EXT.1	A4301
	SP 800-90A	DRBG[8] (HMAC-SHA-256) Prediction Resistance: Enabled	Random Bit Generation for key establishment	FCS_CKM.2 FCS_RBG_EXT.1 FCS_SSHS_EXT.1	A4301
LibMD	FIPS 180-4	SHA-256, SHA-512 (byte Oriented)	Message Digest Generation in password hashing, and in veriexec	FCS_COP.1/Hash FPT_APW_EXT.1 FPT_TST_EXT.1	A4306

[7] Including PKCS#1 v1.5 padding
[8] A Juniper HMAC_DRBG is used in place of the OpenSSL versions of DRBG.

Library	NIST Standard	Algorithm, Mode, Keysize, Function, Hashing, Usage	Cryptographic Operation	SFR(s) supported	CAVP Reference
Kernel	FIPS 180-4	SHA1, SHA-256, SHA-384, SHA-512 (byte Oriented)	Message Digest Generation in verified-exec kernel support	FCS_COP.1/Hash FPT_TST_EXT.1	A4303
	FIPS 198-1	HMAC-SHA1, HMAC-SHA-256 (byte Oriented)	Message Authentication in Kernel provided DRBG	FCS_COP.1/KeyedHash	A4303
	SP 800-90A	DRBG (HMAC-SHA-256) Prediction Resistance: Enabled	Random Bit Generation, provides /dev/random to user applications such as SSH client and server	FCS_RBG_EXT.1	A4303
MACSEC Library	SP 800-38F	AES 128 bits, AES 256 bits	AES Key Wrap	FCS_COP.1/MACSEC	A4304
	SP 800-38B	AES 128 bits, AES 256 bits	AES Keyed Hash Algorithm	FCS_COP.1/CMAC	A4304
	SP 800-90A	DRBG (HMAC-SHA-256)	Random bit generation for key establishment	FCS_RBG_EXT.1 FCS_MACSEC_EXT.3	A4303
MACSEC Acceleration	ISO 19772	AES-GCM (128 bits, 256 bits)	AES encryption and decryption (for BCM82756 PHY)	FCS_COP.1/MACSEC	AES4550
	ISO 19772	AES-GCM (128 bits, 256 bits)	AES encryption and decryption (for BCM54998EM PHY)	FCS_COP.1/MACSEC	C1869
	ISO 19772	AES-GCM (128 bits, 256 bits)	AES encryption and decryption (for BCM84898M PHY)	FCS_COP.1/MACSEC	C1869
	ISO 19772	AES-GCM (128 bits, 256 bits)	AES encryption and decryption (for Juniper Trio 6 PHY)	FCS_COP.1/MACSEC	A4664

Table 12 CAVP Certificate References

43. The FIPS approved algorithms are used when the FIPS mode is enabled[9]. The relevant FIPS knobs are specified in [ECG-304] and [ECG-4100]. (***FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1, FCS_CKM.1, FMT_SMF.1***)

44. 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. The TOE implements Diffie-Hellman group 14, using the modulus and generator specified by Section 3 of RFC3526. (***FCS_CKM.2, FCS_CKM.1***).

45. The TOE acts only as the server for SSH in the supported protocols listed in Table 13:

Protocol	Key Exchange	Authentication	Encryption Algorithms	Data Integrity Algorithms
SSHv2	ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 Diffie-Hellman group 14 (modp 2048)	ssh-rsa rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256 ecdsa-sha2-nistp384 ecdsa-sha2-nistp521	AES CTR 128 AES CTR 256 AES CBC 128 AES CBC 256	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-512

Table 13 Supported SSH Protocols

46. The HMAC algorithms in Table 12 use the values specified in Table 14:

	HMAC-SHA-1	HMAC-SHA-256	HMAC-SHA-512
Key Length	160 bits	256 bits	512 bits
Hash function	SHA-1	SHA-256	SHA-512
Block Size	512 bits	512 bits	1024 bits
Output MAC	160 bits	256 bits	512 bits

Table 14 HMAC Values

47. Junos OS handles zeroization for all CSP, plaintext secret and private cryptographic keys according to Table 15. (***FCS_CKM.4***).

CSP	Description	Method of storage	Storage location	Zeroization 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)	When recommissioned, the config files of the TOE (incl. CSP files such as SSH keys) are removed using the "request vmhost zeroize" option.
	Loaded into memory to complete session establishment	Plaintext	Memory	free() is called by the TOE software at the session termination.
SSH Session Key	Session keys used with SSH, AES 128, 256, hmac-sha-1, hmac-sha2-256 or hmac-sha2-512 key (160, 256 or 512), DH Private Key (2048 or elliptic curve 256/384/521-bits)	Plaintext	Memory	free() is called by the TOE Software at the session termination.

[9] The knob "set system fips chassis level 1" will enforce strict compliance to FIPS and enable restrictions on algorithms and keys sizes as required by FIPS requirements.

CSP	Description	Method of storage	Storage location	Zeroization Method
User Password	Plaintext value entered by user	Plaintext as entered	Processed in Memory	free() is called by the TOE software at the completion of authentication.
		Hashed (HMAC-sha1)	Stored on disk	When the TOE is recommissioned, the config files (including the obfuscated password) are removed using the "request vmhost zeroize" option.
RNG State	Internal state and seed key of RNG	Plaintext	Memory	Handled by kernel, overwritten with zero's at reboot.

Table 15 CSP Storage and Zeroization

48. The CLI implemented by the TOE does not permit the viewing of cryptographic keys. The 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***)

7.1.2 Random Bit Generation

49. The TOE generates random bits in accordance with NIST SP 800-90 using HMAC_DRBG, SHA-256. The RBG in the RE is seeded from the following software sources of entropy:

- RANDOM_INTERRUPT: Hardware devices whose real-time interrupts are known to provide some amount of entropy. The internal representation of handling these interrupts provides entropy. This source can provide entropy both during system boot and steady state.
- RANDOM_NET_ETHER: Timings (CPU counter values at the time of the event) together with the internal representation of network packets are used to harvest entropy that is further fed into the DRBG.
- RANDOM_FS_ATIME. Associated to the time slices during access of the temporary file storage such as a tmpfs. The continuous creation, access and destruction of files in the temporary space in a running system provides randomness. Unpredictability comes from the timing of the time slices.
- RANDOM_ATTACH. Associated with the elapsed cycle count for each device-driver as it attaches to the associated devices in the system and provides entropy during boot-up. Unpredictability comes from the timing of the attachments.

7.1.3 MACsec

50. The TOE implements MACsec for a trusted channel between itself and a peer entity (***FTP_ITC.1/MACSEC***). MACsec is implemented in accordance with IEEE 802.1AE-2006 (***FCS_MACSEC_EXT.1***), supporting:

    a. AES 128/256 ciphersuite (without XPN)

    b. MACsec Key Agreement (MKA) protocol with Static-CAK mode using pre-shared key

    c. Connectivity-Association (CA) per physical port (IFD)

    d. 1 Tx-Secure Channel and 1 Rx- Secure Channel per CA

e. 4 Secure Associations (SA) per SC

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

52. 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 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 PFE traps the packets to RE with ether type 88-8E. (***FCS_MACSEC_EXT.1***)

53. 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***)

54. Each MACsec Key Agreement protocol data unit (MKPDU) transmitted is integrity protected by an 128 bit Integrity Check value (ICV), generated by AES-CMAC using the Integrity Check value Key (ICK). The ICV Key (ICK) is derived from CAK (using AES_CMAC). Before verifying the ICV, the TOE follows Section 11.11.4 of IEEE 802.1X to discard invalid MKPDUs. In particular, it discards MKDPUs whenerver:

    a. the destination address of the MKPDU was an individual address;

    b. the MKPDU is less than 32 octets long;

    c. the MKPDU is not a multiple of 4 octets long;

    d. 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

    e. the CAK Name is not recognized.

    (***FCS_MKA_EXT.1***)

55. The Integrity Check Value (ICV) of MACsec protocol data units (MPDUs) is calculated using the SAK over the destination address, source address, SecTAG, and user data (after encryption, if applicable) and is encoded in the last eight to sixteen octets of theMPDU. The length of the ICV is between 8 and 16 octets, depending on the Cipher Suite. The 64 most significant bits of the 96-bit IV used in generating the ICV are the octets of the SCI and the 32 least significant bits of the 96-bit IV are the octets of the PN. (***FCS_MACSEC_EXT.2***)

56. MACsec allows IPv4/v6 and TCP/UDP headers to be unencrypted while the rest of the frame is encrypted. The offset value for MACsec protected frames are:

    - Offset 0 – Default; Encrypts the entire MPDU payload in the frame
    - Offset 30 – IPv4 & TCP/UDP headers are unencrypted and rest of the payload is encrypted
    - Offset 50 – IPv6 & TCP/UDP headers are unencrypted and rest of the payload is encrypted

57. 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 16. Additionally, the TSF implements a timeout mechanism to ensure a maximum lifetime of two seconds for each MACsec frame (**FPT_DDP_EXT.1.1**).

Table 16 MACsec MKA Timeout values

Timer Use	Timeout (Parameter)	Timeout (Seconds)
Per participant periodic transmission, initialized on each transmission, transmission on expiry	MKA Hello Time MKA Bounded Hello Timeout	2.0-6.0 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.		

58. 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 manager to read it (***FPT_CAK_EXT.1).*** KEK is also derived from CAK. Each participant that considers itself to be the current Key Server can distribute an SAK by encoding the following information in transmitted MKPDUs:

    a. The SAK protected by AES Key Wrap

    b. The Key Number (KN), 32 bits

59. 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

60. SAK is generated using KDF function 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.

61. 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***)

62. 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***)

7.1.4 SSH

63. 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***)

64. 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, FCS_SSHS_EXT.1***)

65. 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, FCS_SSHS_EXT.1***)

66. The SSH server is implemented in accordance with RFCs 4251, 4252, 4253, 4254, 4344, 5656 and 6668. The TOE implements both public key and password-based authentication of administrative users. The conformance to RFCs is given in Table 17.

RFC	Summary	TOE implementation of Security
RFC 4251	The Secure Shell (SSH) Protocol Architecture	**Host Keys:** The TOE uses an ECDSA Host Key for SSH v2, with a key size of 256 bits or greater, which is generated on initial setup of the TOE. It can be de-configured via the CLI and the key will be deleted and thus unavailable during connection establishment. This key is randomly generated to be unique to each TOE instance. The TOE presents the client with its public key and the client matches this key against its known_hosts list of keys. When a client connects to the TOE, the client will be able to determine if the same host key was used in previous connections, or if the key is different (per the SSHv2 protocol). Junos OS also supports RSA-based key establishment schemes with a key size of 2048 bits. **Policy Issues:** The TOE implements all mandatory algorithms and methods. The TOE can be configured to accept public-key based authentication and/or password-based authentication. The TOE does not require multiple authentication mechanisms for users. The TOE allows port forwarding and sessions to clients. The TOE has no X11 libraries or applications and X11 forwarding is prohibited. **Confidentiality:** The TOE does not accept the “none” cipher. Supports AES-CBC-128, AES-CBC-256, AES-CTR-128, AES-CTR-256 encryption algorithms for protection of data over SSH and uses keys generated in accordance with “ssh-rsa”, “rsa-sha2-256”, “rsa-sha2-512”, “ecdsa-sha2-nistp256”, “ecdsa-sha2-nistp384” or “ecdsa-sha2-nistp521” to perform public-key based device authentication. For ciphers whose blocksize >= 16, the TOE rekeys every (2^32-1) bytes. The client may explicitly request a rekeying event as a valid SSHv2message at any time and the TOE will honor this request. Re-keying of SSH session keys can be configured using the sshd_config knob. The data-limit must be between 51200 and 4294967295 (2^32-1) bytes and the time-limit must be between 1 and 1440 minutes. In the evaluated deployment the time-limit must be set within 1 and 60 minutes. **Denial of Service:** When the SSH connection is brought down, the TOE does not attempt to re-establish it. **Ordering of Key Exchange Methods:** Key exchange is performed only using one of the supported key exchange algorithms, which are ordered as follows: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521 (all specified in RFC 5656), diffie-hellman-group14-sha1 (specified in RFC 4253). **Debug Messages:** The TOE sshd server does not support debug messages via the CLI. **End Point Security:** The TOE permits port forwarding. **Proxy Forwarding:** The TOE permits proxy forwarding. **X11 Forwarding:** The TOE does not support X11 forwarding.

RFC 4252	The Secure Shell (SSH) Authentication Protocol	**Authentication Protocol:** The TOE does not accept the "none" authentication method. The TOE implements a timeout period of 30seconds for authentication of the SSHv2 protocol and provides a limit of three failed authentication attempts before sending a disconnect to the client. **Authentication Requests:** The TOE does not accept authentication if the requested service does not exist. The TOE does not allow authentication requests for a non-existent username to succeed – it sends back a disconnect message as it would for failed authentications and hence does not allow enumeration of valid usernames. The TOE denies "none" authentication method and replies with a list of permitted authentication methods. **Public Key Authentication Method:** The TOE supports public key authentication for SSHv2 session authentication. Authentication succeeds if the correct private key is used. The TOE does not require multiple authentications (public key and password) for users. **Password Authentication Method:** The TOE supports password authentication. Expired passwords are not supported and cannot be used for authentication. **Host-Based Authentication:** The TOE does not support the configuration of host-based authentication methods.
RFC 4253	The Secure Shell (SSH) Transport Layer Protocol	**Encryption:** The TOE offers the following for encryption of SSH sessions: aes128-cbc and aes256-cbc, aes128-ctr, aes256-ctr. The TOE permits negotiation of encryption algorithms in each direction. The TOE does not allow the "none" algorithm for encryption. **Maximum Packet length:** Packets greater than 256Kbytes in an SSH transport connection are dropped and the connection is terminated by Junos OS. **Data Integrity:** The TOE permits negotiation of HMAC-SHA1 in each direction for SSH transport. **Key Exchange:** The TOE supports diffie-hellman-group14-sha1. **Key Re-Exchange:** The TOE performs a re-exchange when SSH_MSG_KEXINIT is received.

RFC 4254	Secure Shell (SSH) Connection Protocol	**Multiple channels:** The TOE assigns each channel a number (as detailed in RFC 4251, see above). **Data transfers:** The TOE supports a maximum window size of 256K bytes for data transfer. **Interactive sessions:** The TOE only supports interactive sessions that do NOT involve X11 forwarding. **Forwarded X11 connections:** This is not supported in the TOE. **Environment variable passing:** The TOE only sets variables once the server process has dropped privileges. **Starting shells/commands:** The TOE supports starting one of shell, application program or command (only one request per channel). These will be run in the context of a channel, and will not halt the execution of the protocol stack. **Window dimension change notices:** The TOE will accept notifications of changes to the terminal size (dimensions) from the client. **Port forwarding:** This is fully supported by the TOE.
RFC4344	Secure Shell (SSH) Transport Layer Encryption Modes	**Encryption Modes**: The TOE implements the recommended modes aes128-ctr and aes256-ctr (it does not implement the recommended modes aes192-ctr or 3des-ctr, nor does it implement any of the optional modes).
RFC5656	SSH ECC Algorithm Integration	**ECDH Key Exchange**: The support key exchange methods specified in this RFC are ecdh-sha2-nistp256, ecdh-sha2-nistp384, or ecdh-sha2-nistp521. The client matches the key against its known_hosts list of keys. **Hashing**: Junos OS supports cryptographic hashing via the SHA-256 and SHA-512 algorithms, provided it has a message digest size of either 256 or 512 bits.**Required Curves:** All required curves are implemented: ecdh-sha2-nistp256, ecdh-sha2-nistp384, or ecdh-sha2-nistp521. None of the Recommended Curves are supported as they are not included in [NDcPP2.2E].
RFC 6668	sha2-Transport Layer Protocol	**Data Integrity Algorithms:** Both the recommended and optional algorithms hmac-sha1, hmac-sha2-256 and hmac-sha2-512 (respectively) are implemented for SSH transport.

Table 17 SSH RFC conformance

7.2 Administrator Authentication

67. 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***)

68. 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.

69. 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

70. 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.

71. 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).

72. 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.

73. 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 thie 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.

74. The retry-options can be configured to specify the action to be taken if the administrator fails to enter a valid username/password pair when authenticating from a network management station (***FMT_MTD.1/CoreData***). The retry-options are applied following the first failed login attempt for a given username (***FIA_AFL.1***). The length of delay (5-10 seconds) after each failed attempt is specified by the backoff-factor, and the increase of the delay for each subsequent failed attempt is specified by the backoff-threshold (1-3). The tries-before-disconnect sets the maximum number of times (1-10) the administrator is allowed to enter a password to attempt to log in to the device through SSH before the connection is disconnected. The lockout-period sets the amount of time in minutes before the administrator can attempt to log in to the device after being locked out due to the number of failed login attempts (1-43,200 minutes). Even when an account is locked for remote access to the TOE, an administrator is always able to login locally through the serial console and the administrator can attempt authentication via remote access after the maximum timeout period of 24 hours.

75. 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. (***FIA_UAU_EXT.2***)

76. Passwords are case-sensitive, alphanumeric values. The password has a minimum length of 10 characters and maximum length of 20 characters. It must contain characters from at least two different character sets (upper, lower, numeric, punctuation). Any standard ASCII, extended ASCII and Unicode characters can be selected when choosing a password. (***FIA_PMG_EXT.1***)

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

- The password is hashed when stored using hmac-sha1, sha256 or 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.

78. 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***)

79. 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 logout 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.

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

81. 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.

7.3 Correct Operation

82. 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***):

- Power on test – determines the boot-device responds, and performs a memory size check to confirm the amount of available memory.
- File integrity test – verifies integrity of all mounted signed packages, to assert that system files have not been tampered with. To test the integrity of the software, the fingerprints of the executables and other immutable files are regenerated and validated against the SHA1 fingerprints contains in the manifest file.
- Crypto integrity test – checks integrity of major CSPs, such as SSH hostkeys.
- Authentication error – verifies that veriexec is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real.
- Kernel, libmd, OpenSSL, SSH – verifies correct output from known answer tests for appropriate algorithms.

83. 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.

84. In the event of a transiently corrupt state or failure condition, the system will panic; the event will be logged and the system restarted, having ceased to process network traffic. When the system restarts, the system boot process does not succeed without passing all applicable self-tests. This automatic recovery and self-test behavior, is discussed in Chapter 11 of [ECG-304] and [ECG-4100].

85. When any self-test fails, the device halts in an error state. No command line input or traffic to any interface is processed. The device must be power cycled to attempt to return to operation. This self-test behavior is discussed in [ECG-304] and [ECG-4100] (***FPT_TST_EXT.1***)

7.4 Trusted Update

86. Security Administrators are able to query the current version of the TOE software using the CLI command "show version" (***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,***)

87. 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 7.3. 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 digit signature package verification.

88. 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. (***FCS_COP.1/SigGen, FPT_TUD_EXT.1***)

7.5 Audit

89. The TOE creates and stores audit records for a right set of events. Each event and the content recorded is detailed in Table 9 (***FAU_GEN.1***). Additional auditing for MACsec is generated as stated in Table 10 (***FAU_GEN.1/MACSEC***). Auditing is implemented using syslog.

90. The detail of what events are to be recorded by syslog are determined by the logging level specified the "level" argument of the "`set system syslog`" CLI command. To ensure compliance with the requirements the audit knobs detailed in [ECG-304] and [ECG-4100] must be configured.

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

92. 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

93. 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

94. 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 vmhost zeroize" action is performed and the whole appliance is zeroized (which by definition cannot be recorded).

95. All events recorded by syslog are timestamped. The clock function of Junos OS provides a source of date and time information for the appliance. The clock is used in audit timestamps and maintained using the hardware Time Stamp Counter as the clock source. (***FAU_GEN.2, FPT_STM_EXT.1***)

96. Syslog can be configured to store the audit logs locally (***FAU_STG_EXT.1***). Audit logs can also be sent to one or more syslog log servers in real time via Netconf over SSH. The sending of the audit logs is done automatically without Administrator intervention (***FAU_STG.1, FMT_MOF.1/Functions***).

97. Local audit log are stored in /var/log/ in the filesystem. Only a Security Administrator can read or delete log 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 user, using the "`size`" argument for the "`set system syslog`" CLI command.

98. 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.

99. 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.

7.6 Management

100. Accounts assigned to the Security Administrator role are used to manage Junos OS in accordance with [NDcPP2.2E]. 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 [NDcPP2.2E]. (***FMT_SMR.2***)

101. The TOE allows user access either through the system console or remotely over SSH. Users are required to provide unique identification and authentication data before any access to the system is granted, as detailed in Sect. 7.2. (***FMT_SMR.2, FMT_SMF.1***)

102. The Security Administrator has the capability to:

- Administer the TOE locally via the serial ports on the physical device or remotely over an SSH connection.
- Initiate a manual update of TOE software (***FMT_MOF.1/ManualUpdate***):
  - Query currently executing version of TOE software (***FPT_TUD_EXT.1***)
  - Verify update using digital signature (***FPT_TUD_EXT.1***)
- Manage Functions:
  - Transmission of audit data to an external IT entity, including 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***)
  - Handling of audit data, including setting limits of log file size (***FMT_MOF.1/Functions***)
- Manage TSF data (***FMT_MTD.1/CoreData***)
  - Create, modify, delete administrator accounts, including configuration of authentication failure parameters
  - Reset administrator passwords
  - Re-enable an Administrator account (***FIA_AFL.1***);
- Manage crypto keys as described in Table 18. (***FMT_MTD.1/CryptoKeys***)

Table 18 Cryptographic Keys and Parameters

Name	Generation	Import/ Export	Establishment	Storage	Zeroisation	Use & related keys
SSH-PUB	*SSH-KeyGen tool (Uses OpenSSL EC_KEY_generate_key and rsa_builtin_keygen API) Random Number generation Alg: HMAC DRBG*	*Entry: N/A Output: Plaintext during SSH session establishment*	*N/A*	*Plaintext: Persistent*	*Zeroize Command*	*SSH-2 Public Host Key: 1st time SSH-2 is configured the ECDSA and RSA keys are generated.* *Used to Identify the host.*
Auth-User Pub	*Externally generated*	*Entry: Manual entry*	*N/A*	*Plaintext: Persistent*	*Zeroize Command*	*User Authentication Public Keys: ECDSA and RSA public keys* *Used to authenticate users to the module.*
Auth-CO Pub	*Externally generated*	*Entry: Manual*	*N/A*	*Plaintext: Persistent*	*Zeroize Command*	*CO Authentication Public Keys: ECDSA P256, P-384* *Used to authenticate users to the module.*
Root-CA	*Externally generated*	*Loaded at manufacture time.*	*N/A*	*Persistent*	*Not zeroized*	*ECDSA prime256v1 X.509 V3 or prime384v1 X.509 V3 Certificate* *Used to verify the validity of the Package-CA.*
Package-CA	*Externally generated*	*Loaded at manufacture time.*	*N/A*	*Plaintext: Persistent*	*Not zeroized*	*ECDSA prime256v1 X.509 V3 Certificate* *Certificate that holds the public key of the signing key that was used to generate all the signatures used on the packages and signatures lists.*

SSH-DH-PUB	*Internally generated using HMAC DRBG (part of DH exchange)*	*N/A*	*SP800-56Arev3 compliant KAS DH groups 14, 19,20 and 21*	*Plaintext: RAM*	*Reboot & session termination*	*SSH DH and ECDH Public Keys* *Used with SSH-2 for key establishment:*

- Perform management functions (***FMT_SMF.1***):
  - Configure the access banner (***FTA_TAB.1***)
  - Configure the session inactivity time before session termination or locking, including termination of session when serial console cable is disconnected (***FTA_SSL_EXT.1, FTA_SSL.3***)
  - Manage cryptographic functionality (***FCS_SSHS_EXT.1***), including:
    - ssh ciphers
    - hostkey algorithm
    - key exchange algorithm
    - hashed message authentication code
    - thresholds for SSH rekeying
  - Perform MACsec management functions (***FMT_SMF.1/MACSEC***):
    - Ability to generate a PSK and install it in the device
    - CLI commands to manage the Key Server to create, delete, and activate MKA participants
    - Enable, disable, or delete a PSK-based CAK using CLI commands
  - Set the system time (***FPT_STM_EXT.1***)

103. Detailed topics on the secure management of Junos OS are discussed in [ECG-304] and [ECG-4100].

8 Glossary

AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Program Interface
cPP	collaborative Protection Profile
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFP	C Form-factor Pluggable
CSP	Critical security parameter
DH	Diffie Hellman
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EP	Extended Package, defined in [CC1]
ESP	Encapsulating Security Payload
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Authentication Code
I&A	Identification and Authentication
ID	Identification
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv6	Internet Protocol Version 6
ISO	International Organization for Standardization
IT	Information Technology
Junos	Juniper Operating System
MIC	Modular Interface Cards
MPC	Modular Port Concentrator
MS-MPC	MultiServices Modular Port Concentrator
NAT	Network Address Translation
NDcPP	Network Device collaborative Protection Profile
NTP	Network Time Protocol
OSI	Open Systems Interconnect
OSP	Organizational Security Policy
PAM	Pluggable Authentication Module
PFE	Packet Forwarding Engine
PIC/PIM	Physical Interface Card/Module
PKI	Public Key Infrastructure
POE	Power over Ethernet
PP	Protection Profile
PRNG	Pseudo Random Number Generator
RE	Routing Engine
RFC	Request for Comment
RNG	Random Number Generator
RSA	Rivest, Shamir, Adelman
SA	Security Association
SFP	Small Form-factor Pluggable
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SSH	Secure Shell

SSL	Secure Sockets Layer
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF interfaces
UDP	User Datagram Protocol