Security Target Junos OS 22.3R1 for ACX5448-M, EX4400-24T, EX4400-24P, EX4400-24MP, EX4400-48T, EX4400-48P, EX4400-48F, EX4400-48MP and QFX5120-48YM

Juniper Networks, Inc.

Version 1.0

29 February, 2024

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

Abstract

This document is a Security Target (ST) which provides the basis for an evaluation of a specific Target of Evaluation (TOE), Junos OS 22.3R1 for ACX5448-M, EX4400-24T, EX4400-24P, EX4400-24MP, EX4400-48T, EX4400-48P, EX4400-48F, EX4400-48MP and QFX5120-YM. This ST is conformant to the requirements of Collaborative Protection Profile for Network Devices v2.2E and MACsec Ethernet Encryption Extended package v1.2.
The evaluation of the TOE is carried out in accoradance with Common Criteria v3.1 Revision 5 defined in [CC1], [CC2] and [CC3] as well as the Common Evaluation Methodology v3.1 Revision 5 [CEM] and the Supporting Document for the Collaborative Protection Profile for Network Devices v2.2E [SD].

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-013-v2.0, Version 2.0, 30-Sep-2021.

[MACsec] Network Device Collaborative Protection Profile (NDcPP) Extended Package MACsec Ethernet Encryption, May 10, 2016, version 1.2

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

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

Table of Contents

1 ST 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 and TOE Reference

ST Title	Security Target Junos OS 22.3R1 for ACX5448-M, EX4400-24T, EX4400-24P, EX4400-24MP, EX4400-48T, EX4400-48P, EX4400-48F, EX4400-48MP and QFX5120-48YM
ST Revision	1.0
ST Draft Date	29 February, 2024
Author	Juniper Networks, Inc.
cPP/EP Conformance	[NDcPP], [MACsec]
TOE Title	Junos OS 22.3R1 for ACX5448-M, EX4400-24T, EX4400-24P, EX4400-24MP, EX4400-48T, EX4400-48P, EX4400-48F, EX4400-48MP and QFX5120-YM
TOE Firmware	Junos OS 22.3R1

1.2 Document Structure

2. This Security Target follows the format summarized in Table 1.

Table 1 Document Organization

Section	Title	Description
1	Introduction	An overview of the TOE and defines the hardware and firmware that make up the TOE as well as the physical and logical boundaries of the TOE
2	Conformance Claims	States the conformance to Common Criteria versions, Protection Profiles, and Packages where applicable
3	Security Problem Definition	States the threats, assumptions and OSPs that affect the TOE
4	Security Objectives	States the security objectives for the TOE and for the operational environment and a rationale to demonstrate that the security objectives satisfy the threats
5	Security Functional Requirements	A statement of the Security Functional Requirements for the TOE
6	Security Assurance Requirements	A statement of the Security Assurance Requirements for the TOE
7	TOE Summary Specification	Identifies the IT security functions implemented by the TOE and also identifies the assurance measures which meet the security assurance requirements
8	Glossary	Identifies the terminology used in the ST.

1.3 Document Conventions

3. This document follows the same conventions as those applied in [NDcPP] and [MACsec] 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 (ECD);
- Refinement made in the ST: the refinement text is indicated with **bold text** and ~~strikethroughs~~;
- Selection completed in the ST: the selectiogn values are indicated with <u>underlined text</u>

e.g. "[*selection: disclosure, modification, loss of use*]" in [CC2] or an ECD might become "disclosure" (completion;

- 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");
- Any other formatting conventions of the statement of Security Functional Requirements in [NDcPP] and [MACsec] are maintained.

1.4 TOE Overview

4. The Target of Evaluation (TOE) is Juniper Networks, Inc. Junos OS 22.3R1 executing on specific ACX5000, EX4400 and QFX5000 Series Ethernet Switches with MACsec. Each TOE model is a specific network appliance implementing the same functionality on a different hardware platform.
5. The following appliance models constitute the variations of the TOE:
   - ACX5448-M with Junos OS 22.3R1 software, offering 44 1GbE/10GbE and six 40GbE/100GbE access ports;
   - EX4400-24T with Junos OS 22.3R1 software, offering 24 x 1GbE non-PoE access ports;
   - EX4400-24P with Junos OS 22.3R1 software, offering 24 x 1GbE PoE access ports, delivering up to 90 W per port with an overall total 1440 W of PoE power budget (using two power supplies);
   - EX4400-48T with Junos OS 22.3R1 software, offering 48 x 1GbE non PoE-access ports;
   - EX4400-48P with Junos OS 22.3R1 software, offering 48 x 1GbE PoE access ports, delivering up to 90 W per port with an overall total 1800 W of PoE power budget (using two power supplies);
   - EX4400-48F with Junos OS 22.3R1 software, offering 12 x 10GbE SFP+ and 36 x 1GbE SFP fiber access ports; and
   - QFX5120-YM with Junos OS 22.3R1 software, offering 48 25GbE (SFP28)/10GbE (SFP+)/1GbE (SFP) ports.
6. The appliance variations constituting the TOE are secure network devices that protect themelves largely by offering only a minimal logical interface to the network and attached nodes. They include the Junos OS firmware, Junos OS 22.3R1, which is a special purpose OS offering no general purpose computing capabilities. Junos OS implements both management and control functions as well as all IP routing.
7. The appliances primarily function is to support the definition and enforcement of information flow policies among network nodes. Each information flow from one network node to another passes through an instance of the TOE. Information flow is controlled on the basis of network node addresses and protocol. The TOE also ensures that security-relevant activity is audited and implemenets the necessary tools to manage all security functions.

1.5 TOE Description

1.5.1 Overview

8. The variants of the TOE share a common architecture and feature set. They implement a variety of high-speed interfaces (only Ethernet is in the scope of the evaluation) for enterprise branch, campus, and data center networks. They also share common Junos firmware, features, and technology for compatibility across platforms.

9. The appliances are physically self-contained. Each appliance houses the firmware and hardware necessary to perform all routing functions. The architecture components of the appliances are:

   - Switch fabric – the switch fabric boards/modules provide a highly scalable, non-blocking, centralized switch fabric matrix through which all network data passes.
   - Routing Engine (Control Board) – the Routine Engine (RE) runs the Junos firmware and implements Layer 3 routing services and Layer 2 switching services. The RE also implements the management functions for configuration and operation of the TOE and controls the flow of information through the TOE, including support for appliance interface control and control plane functions such as chassis component, system management and user access to the appliance.
   - Layer 2 switching services, Layer 3 switching/routing services and network management for all operations necessary for the configuration and operation of the TOE and controls the flow of information through the TOE.
   - Packet Forwarding Engine (PFE) – The PFE implements all operations necessary for transit packet forwarding. The PFE implements an extensive set of Layer 2 and Layer 3 services that can be deployed in any combination of L2- L3 applications.
   - Power – The ACX5448-M, EX4400-24T, EX4400-48T, EX4400-48F, and QFX5120-48YM variants of the TOE include non-PoE ports. The EX4400-24P and EX4400-48P variants implement a Power over Ethernet (PoE) interface for powering up the TOE. The EX4400-24MP and EX4400-48MP variants support both options. Power supply bays allow flexibility for provisioning and redundancy. The power supplies connect to the midplane, which distributes the different output voltages produced by the power supplies to the appliance components, depending on their voltage requirements.

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

11. The appliances support numerous routing and switching standards for flexibility and scalability.

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

13. The TOE implements MACsec between adjacent devices. All traffic communicated between the devices including frames for LLDP, DHCP, ARP, STP, Ethernet Control frames, etc (the exceptions to this protection are Destination MAC and Source MAC addresses in MACsec and MKA frames).

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

1.5.2 Physical Boundary

15. The TOE is an appliance consisting of the Junos OS 22.3R1 for ACX5448-M, EX4400-24T, EX4400-24P, EX4400-24MP, EX4400-48T, EX4400-48P, EX4400-48F, EX4400-48MP and QFX5120-YM appliance chasses. Hence, the TOE is contained within the physical boundary of the specified appliance chassis, as shown in Figure 1. The physical boundary of the TOE is the entire chassis of the appliance.

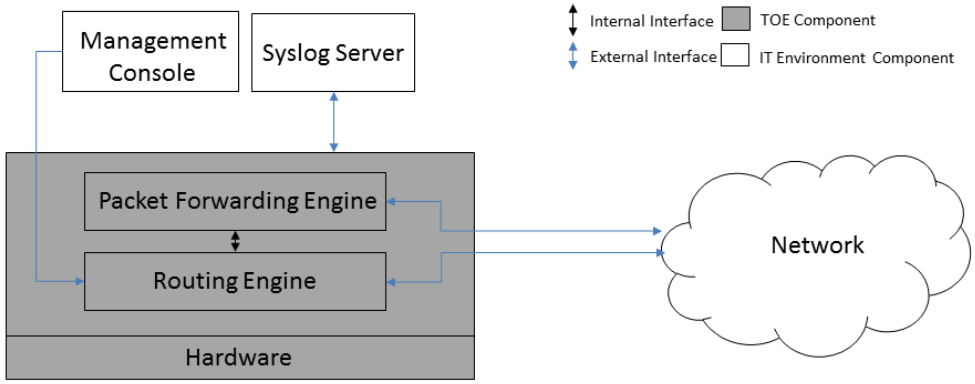

Figure 1 TOE Boundary

16. The TOE interfaces are comprised of the network interfaces which pass traffic, and the management interface which handle administrative actions.

17. The jinstall images for the TOE are the following:

    a. For ACX5448-M: junos-vmhost-install-acx-x86-64-22.3R1.11.tgz

    b. For the EX4400 variants: junos-install-ex-x86-64-22.3R1.11.tgz

    c. For QFX5120-48YM: jinstall-host9-qfx-5e-ng-x86-64-22.3R1.11-secure-signed.tgz

18. The firmware version reflects the detail reported for the components of the Junos OS when the “show version” command is executed on the appliance.

19. The guidance documents included as part of the TOE are:

    [ECG1] Common Criteria Evaluated Configuration Guide for EX4400-24MP, EX4400-24P, EX4400-24T, EX4400-48F, EX4400-48MP, EX4400-48P, and EX4400-48T Devices. Published 23-02-2023, Release 22.3R1

[ECG2] Common Criteria Configuration Guide for ACX5448 Devices. Published 28-02-2023, Release 22.3R1.

[ECG3] Common Criteria Evaluated Configuration Guide for EX4650-48Y, QFX5120-32C, QFX5120-48T, QFX5120-48Y, and QFX5120-48YM Devices. Published 23-02-2023, Release 22.3R1.

1.5.3 Logical Scope of the TOE

The logical scope of the TOE includes the security functionality stated in Table 2.

Table 2 Logical Scope of the TOE

Security Functionality	Description
Security Audit	Auditable events are stored in the syslog files on the appliance and can be sent to an external log server (via Netconf over SSH). Auditable events include start-up and shutdown of the audit functions, authentication events, and all SFR-specific events required by the applicable Protection Profiles. Audit records include the date and time, event category, event type, username, and the outcome of the event (success or failure). Local syslog storage limits are configurable and are monitored. If the storage limit is reached the oldest logs will be overwritten.
Cryptographic Support	The TOE implements an SSH server to support protected communications for administrators to establish secure sessions and to connect to external syslog servers. The TOE requires that applications exchanging information with it are successfully authenticated prior to any exchange (i.e. applications connecting over SSH). Communication over point-to-point links between Juniper appliances can be secured using MACsec. The TOE includes cryptographic modules that implement the underlying cryptographic services, including key management and protection of stored keys, algorithms, random bit generation and crypto-administration. The cryptographic modules provide confidentiality and integrity services for authentication and for protecting communications with connecting applications.
Identification and Authentication	The TOE supports Role Based Access Control. All users must be authenticated to the TOE prior to being granted access to any management actions. The TOE supports password based authentication and public key based authentication. Based on the assigned role, a user is granted a set of privileges to access the system. Administrative users must provide unique identification and authentication data before any administrative access to the system is granted. Authentication data entered and stored on the TOE is protected.
Security Management	The TOE provides a Security Administrator role that is responsible for:

	• configuration and maintenance of cryptographic elements related to the establishment of secure connections to and from the evaluated product • regular review of all audit data; • initiation of trusted update function; • administration of MACsec functionality; • all administrative tasks (e.g., creating the security policy). The devices are managed through a Command Line Interface (CLI). The CLI is accessible through local (serial) console connection or remote administrative (SSH) session.
Protection of the TSF	The TOE protects all passwords, pre-shared keys, symmetric keys and private keys from unauthorized disclosure. Passwords are stored using sha256 or sha512. The TOE executes self-tests during initial start-up to ensure correct operation and enforcement of its security functions. An administrator can install software updates to the TOE. The TOE internally maintains the date and time.
TOE Access	Prior to establishing an administration session with the TOE, a banner is displayed to the user. The banner messaging is customizable. The TOE will terminate an interactive session after a period of inactivity. A user can terminate their local CLI session and remote CLI session by entering exit at the prompt.
Trusted Path/Trusted Channel	The TOE supports SSH v2 for secure communication to Syslog server. The TOE supports SSH v2 (remote CLI) for secure remote administration.

1.5.4 Non-TOE Hardware/Software/Firmware

The TOE relies on the provision of the following items in the network environment:

- Syslog server supporting SSHv2 connections to send audit logs;
- SSHv2 client for remote administration;
- Serial connection client for local administration.

1.5.5 Summary of Out of 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 Claims

2.1 CC Conformance Claim

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

This TOE is 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

This TOE is conformant to:

- [NDcPP] Collaborative Protection Profile for Network Devices (NDcPP), Version 2.2E, 27 March 2020
- [MACsec] Network Device collaborative Protection Profile (NDcPP) Extended Package MACsec Ethernet Encryption (MACSECEP), Version 1.2, 10 May 2016

There is no applicable PP-Configuration in accordance to which the conformance to the above should be claimed.

2.3 Conformance Rationale

This Security Target provides exact conformance to Version 2.2E of the Collaborative Protection Profile for Network Devices and to version 1.2 of the NDcPP Extended Package MACsec Ethernet Encryption [MACsec]. The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile and extended package performing only operations defined there.

2.4 Technical Decisions

All NIAP Technical Decisions relevant to [NDcPP] and [MACsec] and their applicability to the TOE is given in Table 3.

Table 3 Technical Decisions

NIAP Technical Decisions (TDs)		
Technical Decisions	**Applicable**	**Exclusion Rationale (if applicable)**
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 implements neither DTLS client nor TLS client.
TD0738: NIT Technical Decision for Link to Allowed-With List	Yes	
TD0670: NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	No	The TOE does not include TLS client functionality

TD0654: MACsec data delay protection and updated conditional support for group CAK	Yes	
TD0652: MACsec CAK Lifetime in FMT_SMF.1	Yes	
TD0639: NIT Technical Decision for Clarification for NTP MAC Keys	No	The TOE does not include NTP
TD0638: NIT Technical Decision for Key Pair Generation for Authentication	Yes	
TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No	The TOE does not include a SSH client in scope.
TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters	No	The TOE does not include a TLS server in scope
TD0634: NIT Technical Decision for Clarification required for testing IPv6	No	Superseded by TD0790.
TD0633 NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	Yes	
TD0632 NIT Technical Decision for Consistency with Time Data for vNDs	No	The TOE is not a virtual TOE.
TD0631 NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes	
TD0618: MACsec Key Agreement and conditional support for group CAK	Yes	
TD0592: NIT Technical Decision for Local Storage of Audit Records	Yes	
TD0591: NIT Technical Decision for Virtual TOEs and hypervisors	No	The TOE is not a virtual TOE.
TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
TD0572: NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
TD0571: NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
TD0570: NiT Technical Decision for Clarification about FIA_AFL.1	Yes	
TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	The TOE does not claim TLS or DTLS
TD0564: NiT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
TD0563: NiT Technical Decision for Clarification of audit date information	Yes	
TD0556: NIT Technical Decision for RFC 5077 question	No	The TOE does not claim TLS
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	The TOE does not claim TLS

TD0553: FCS_MACSEC_EXT.1.4 and MAC control frames	Yes	
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
TD0546: NIT Technical Decision for DTLS - clarification of Application Note 63	No	The TOE does not claim DTLS
TD0538: NIT Technical Decision for Outdated link to allowed-with list	No	Superseded by TD0738
TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	No	The TOE does not claim TLS
TD0536: NIT Technical Decision for Update Verification Inconsistency	Yes	
TD0535: NIT Technical Decision for Clarification about digital signature algorithms for FTP_TUD.1	Yes	
TD0533: NIT Technical Decision for FTP_ITC.1 with signed downloads	Yes	
TD 0532: NIT Technical Decision for Use of seeds with higher entropy	Yes	
TD0531: NIT Technical Decision for Challenge-Response for Authentication	Yes	
TD0530: NIT Technical Decision for FCS_TLSC_EXT.1.1 5e test clarification	No	No FCS_TLSC_EXT.1 claimed
TD0512: Group CAKs for establishing multiple MKA connections is not mandated	No	Superseded by TD0652
TD0529: NIT Technical Decision for OCSP and Authority Information Access extension	No	TOE does not claim certificate authentication of firmware updates
TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No	FCS_NTP_EXT.1.4 not claimed
TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes	No X509 support is claimed
TD0509: Correction to MACsec Audit	Yes	
TD0487: Correction to Typo in FCS_MACSEC_EXT.4	Yes	
TD0466: Selectable Key Sizes for AES Data Encryption/Decryption	Yes	
TD0273: Rekey after CAK expiration	Yes	
TD0190: FPT_FLS.1(2)/SelfTest Failure with Preservation of Secure State and Modular Network Devices	Yes	
TD0135: SNMP in NDcPP MACsec EP v1.2	No	No SNMP claimed
TD0105: MACsec Key Agreement	Yes	

3 Security Problem Definition

As this TOE is neither a distributed nor a virtual Network Device, none of the threats/assumptions/OSPs relating to distributed or virtual Network Device TOEs are applicable to this TOE.

3.1 Threats

The following threats for this TOE are as defined in [NDcPP] Section 4.1, which also apply to [MACsec]. Namely:

- 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 includes 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.

The following additional threats specified in [MACsec] are also detailed for this TOE:

- 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.

- T.UNTRUSTED_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.

- T.DATA_INTEGRITY

  An attacker may modify data transmitted over the MACsec channel in a way that is not detected by the recipient.

3.2 Assumptions

This section describes the assumptions made in identification of the threats and security requirements for network devices. The network device is not expected to provide assurance in any of these areas, and as a result, requirements are not included to mitigate the threats associated. The assumptions made for this TOE are as defined in [NDcPP].

- A.PHYSICAL_PROTECTION

  The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows

unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.

- A.LIMITED_FUNCTIONALITY

  The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).

- A.TRUSTED_ADMINSTRATOR

  The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.

- A.REGULAR_UPDATES

  The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

- A.ADMIN_CREDENTIALS_SECURE

  The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.

- A.RESIDUAL_INFORMATION

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

The following assumption has been updated as per [MACsec]:

- A.NO_THRU_TRAFFIC_PROTECTION

  This assumption is only applicable to interfaces in the TOE that are defined by the [NDcPP]. For these interfaces, the TOE does not provide any assurance regarding the protection of traffic that traverses it.

3.3 Organizational Security Policies

An organizational security policy (OSP) is a set of rules, practices, and procedures imposed by an organization to address its security needs. There is one single policy applied to this TOE, as defined in [NDcPP] Section 4.3.

- 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.

4 Security Objectives

The security objectives have been taken from [NDcPP] and are reproduced here for the convenience of the reader. As this TOE is not distributed nor a virtual Network Device, none of the obejectives relating to distributed TOEs or virtual Network Devices are specified for this TOE.

4.1 Security Objectives for the TOE

The security objectives for the TOE are trivially determined through the inverse of the statement of threats presented in [NDcPP] Section 4.1.

These are augmented by the statement of security objectives for the TOE in relation to the MACsec capabilities as detailed in [MACsec] Section 3, namely:

- O.CRYPTOGRAPHIC_FUNCTIONS

  The TOE will provide cryptographic functions that are used to establish secure communications channels between the TOE and the Operational Environment.

- O.AUTHENTICATION

  The TOE will provide the ability to establish connectivity associations with other MACsec peers.

- O.PORT_FILTERING

  The TOE will provide the ability to restrict the flow of traffic between networks based on originating port and established connection information.

- O.SYSTEM_MONITORING

  The TOE will provide the means to detect when security-relevant events occur and generate audit events in response to this detection.

- O.AUTHORIZED_ADMINISTRATION

  The TOE will provide management functions that can be used to securely manage the TSF.

- O.TSF_INTEGRITY

  The TOE will provide mechanisms to ensure that it only operates when its integrity is verified.

- O.REPLAY_DETECTION

  The TOE will provide the means to detect attempted replay of MACsec traffic by inspection of packet header information.

- O.VERIFIABLE_UPDATES

  The TOE will provide a mechanism to verify the authenticity and integrity of product updates before they are applied.

4.2 Security Objectives for the Operational Environment

The statement of security objectives for the operational environment of this TOE is as defined in [NDcPP] Section 5.1, with the exception of with the exception of OE.NO_THRU_TRAFFIC_PROTECTION, whose scope is modified as per [MACsec] Section 3.2.

- 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.TRUSTED_ADMIN

  Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

- 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.

- OE.NO_THRU_TRAFFIC_PROTECTION

  Except for interfaces covered by the [MACsec], 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.

4.3 Security Objectives Rationale

As these objectives for the TOE and operational environment are the same as those specified in [NDcPP] and [MACsec], the rationales provided in the prose of the following are wholly applicable to this security target as the statements of threats, assumptions, OSPs and security objectives provided in this security target are the same as those defined in the collaborative Protection Profile and Extended Package to which this ST claims conformance:

- [NDcPP] Section 4
- [MACsec] Section 3 and Appendix A

5 Security Functional Requirements

All security functional requirements are taken from the [NDcPP] and [MACsec]. The Security Functional requirements are primarily structured according to [NDcPP], with requirements and operations from [MACsec] inserted as appropriate. The SFRs are presented in accordance with the conventions described in [NDcPP] Section 6.1, and Section 1.4 of this document.

Note: as this TOE is not distributed nor a virtual Network Device, none of the security functional requirements from [NDcPP] relating to distributed TOEs and virtual Network devices are specified for this TOE.

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[1]

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 4*.*

ST Application Note:

The "Services" referenced in the above requirement relate to the trusted communication channel to the external syslog server (netconf over SSH) and and the trusted path for remote administrative sessions (SSH, which can be tunneled over IPsec).

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 4.

Table 4 Auditable Events

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

[1] The list of auditable events in **Error! Reference source not found.** is a superset of all those specified in REF NDcPP \h * MERGEFORMAT [NDcPP] and [MACsec], incorporating TD0509.

FAU_GEN.2	None	None
FAU_STG_EXT.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_COP.1(1)/KeyedHashCMAC	None.	None.
FCS_COP.1/MACsec	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.
FCS_MACSEC_EXT.1	Session establishment	Secure Channel Identifier (SCI)
FCS_MACSEC_EXT.4.4	Creation of Connectivity Association	Connectivity Association Key Names
FCS_MACSEC_EXT.3.1	Creation and update of Secure Association Key	Creation and update times
FIA_AFL.1	Administrator lockout due to excessive authentication failures	None
FPT_RPL.1	Detected replay attempt	None

5.1.1.2 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.

ST Application Note
Transfer of the audit data to the external server is performed automatically (without further Security Administrator intervention) in the evaluated deployment.

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

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

FAU_STG_EXT.1.3 The TSF shall [overwrite previous audit records according to the following rule: *[oldest log is overwritten]*] when the local storage space for audit data is full.

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[2]

FCS_CKM.2.1 The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: *[*

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

[2] Incorporates TD0581 and TD0580.

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 [GCM, 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, GCM as specified in ISO 19772]*.

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, 4096 bits],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [P-256, P-384, P-521 bits]*

] ~~and cryptographic key sizes [assignment: *cryptographic key sizes*]~~

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, 384 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(1)/KeyedHashCMAC Cryptographic Operation (AES-CMAC Keyed Hash Algorithm))[3]

FCS_COP.1.1(1)/KeyedHash: CMAC Refinement: 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 NIST SP 800-38B.

[3] Incorporates TD0466

FCS_COP.1(5) Cryptographic Operation (MACsec AES Data Encryption/Decryption)[4]

FCS_COP.1.1(5) Refinement The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in AES Key Wrap, GCM and cryptographic key sizes [128 bits, 256 bits] that meet 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.

5.2.3 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, FCS_MACSEC, FCS_MKA)

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 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 [*263K*] 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*][5].

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).

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 each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

[4] Incorporates TD0466

5.2.4.2 FCS_MACSEC_EXT.1 MACsec

FCS_MACSEC_EXT.1 MACsec[6]

FCS_MACSEC_EXT.1.1 The TSF shall implement MACsec in accordance with IEEE Standard 802.1AE-2006.

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 a MACsec Protocol Data Unit (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 (PAE EtherType 88-8E) and MACsec frames (EtherType 88-E5) and control frames (EtherType is 88-08) and shall discard others.

5.2.4.3 FCS_MACSEC_EXT.2 MACsec Integrity and Confidentiality

FCS_MACSEC_EXT.2 MACsec Integrity and Confidentiality [7]

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 Secure Association Key (SAK).

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

5.2.4.4 FCS_MACSEC_EXT.3 MACsec Randomness

FCS_MACSEC_EXT.3 MACsec Randomness [8]

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.

ST Application Note:
As part of the key derivation a nonce from the TOE's random bit generator is used as one of the inputs, but the CAK is generated in accordance with section 9.8.1 of IEEE 802.1X-2010.

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

5.2.4.5 FCS_MACSEC_EXT.4 MACsec Key Usage

FCS_MACSEC_EXT.4 Key Usage[9]

FCS_MACSEC_EXT.4.1 The TSF shall support peer authentication using pre-shared keys, [no other methods].

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

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

[6] Specified in [MACsec]. Incorporates TD0553.
[7] Specified in [MACsec].
[8] Specified in [MACsec].
[9] Specified in [MACsec]. Incorporates TD0487.

FCS_MACSEC_EXT.4.4 The TSF shall associate Connectivity Association Key Name (CKN) with Security Association Key (SAK)s that are defined by the key derivation function using the CAK as input data (per 802.1X, section 9.8.1).

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

5.2.4.6 FCS_MKA_EXT.1 MACsec Key Agreement

FCS_MKA_EXT.1 MACsec Key Agreement[10]

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 enable data delay protection for MKA that ensures MKA data frames are not delayed by more than 2 seconds.

FCS_MKA_EXT.1.3 The TSF shall provide assurance of the integrity of MKA protocol data units (MKPDUs) using an Integrity Check Value (ICV) derived from an Integrity Check Value Key (ICK).

FCS_MKA_EXT.1.4 The TSF shall provide the ability to derive an Integrity Check Value Key (ICK) from a CAK using a KDF.

FCS_MKA_EXT.1.5 The TSF shall enforce an MKA Lifetime Timeout limit of 6.0 seconds and MKA Bounded Hello Time limit of 0.5 seconds.

Application Note: The Key Server may distribute a group CAK established by pairwise CAKs.

FCS_MKA_EXT.1.6 The Key Server shall refresh a SAK when it expires. The Key Server shall distribute a SAK by [pairwise CAKs]. ~~If group CAK is selected, then the Key Server shall distribute a group CAK by [selection: a group CAK, pairwise CAKs, pre-shared key].~~ If pairwise CAK is selected, then the pairwise CAK shall be [pre-shared key]. The Key Server shall refresh a CAK when it expires.

FCS_MKA_EXT.1.7 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.1.8 The TSF shall validate MKPDUs according to 802.1X, 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 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.
e) 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 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 802.1X, section 9.4.1 shall be decoded as specified in 802.1X, section 11.11.4.

[10] Specified in [MACsec]. Incorporates TD0105 and TD0654.

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 using a password.*

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

ST Application Note
The Security Administrator can select to unlock the account of another administrator who has failed to authenticate, rather than require the administrator to wait until the delay of an administrator-configured time period has lapsed before another attempt can be made to authenticate.

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;
- [*[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 enable and disable 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[11]

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

- *Ability to administer the TOE locally and remotely;*

[11] Incorporates TD0652

- *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;*
- *Generate a PSK and install it in the device ([MACsec])*
- *Manage the Key Server to create, delete, and activate MKA participants [[other management function – CLI commands]] ([MACsec])*
- *Specify a lifetime of a CAK([MACsec])*
- *Enable, disable, or delete a PSK in the CAK cache of a device using [[other management function– CLI commands]] ([MACsec])*
- *Configure the number of failed administrator authentication attempts that will cause an account to be locked out*

[

- *Ability to configure audit behaviour;*
- *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 configure the reference identifier for the peer].*

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 administrative passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext administrative 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*[12]].

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

5.5.6 Protection of CAK Data (FPT_CAK_EXT.1)

5.5.6.1 FPT_CAK_EXT.1 Protection of CAK Data

5.5.6.1 FPT_CAK_EXT.1 Protection of CAK Data [13]

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

[12] The complete list of algorithm tests is provided in [ECG1][ECG2][ECG3] “Performing Self-Tests on a Device”.
[13] Specified in [MACsec].

5.5.7 Self-Test Failures (FPT_FLS)

5.5.7.1 FPT_FLS.1/SelfTest Fail Secure with Preservation of Secure State

5.5.6.1 FPT_FLS.1(2)/SelfTest Fail Secure with Preservation of Secure State[14]

FPT_FLS.1.1(2)/SelfTest Refinement: The TSF shall **shut down** when any of the following types of failures occur: **failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests**.

5.5.8 Replay Detection (FPT_RPL.1)

5.5.8.1 FPT_RPL.1 Replay Detection

5.5.6.1 FPT_RPL.1 Replay Detection[15]

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

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

5.6 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.

[14] Specified in [MACsec]. Incorporates TD0190.
[15] Specified in [MACsec].

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

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, MACsec] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [*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 **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 [*MACsec communication*].

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 provides confidentiality and integrity, 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*.

5.7.3 TOE Security Functional Requirements Rationale

The SFRs for the TOE are the same as those specified in [NDcPP] and [MACsec]. The rationale for the SFRs taken from [NDcPP] is provided in the Section 4.1 of [NDcPP]. The rationale for the SFRs taken from [MACsec] is given in Sect. 3 of [MACsec].

6 Security Assurance Requirements

The TOE security assurance requirements are taken from [NDcPP] Section 7, as listed in Table 5. There are no additional security assurance requirements stated in [MACsec].

Table 5 Security Assurance Requirements

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)

7 TOE Summary Specification

7.1 Security Audit

Junos OS creates and stores audit records for the following events (the detail of content recorded for each audit event is detailed in Table 10 (***FAU_GEN.1***). Auditing is implemented using syslog.

- Start-up and shut-down of the audit functions
- Administrative login and logout
- Configuration is committed
- Configuration is changed (includes all management activities of TSF data)
- Generating/import of, changing, or deleting of cryptographic keys (see below for more detail)
- Resetting passwords
- Starting and stopping services
- All use of the identification and authentication mechanisms
- Unsuccessful login attempts limit is met or exceeded
- Any attempt to initiate a manual update
- Result of the update attempt (success or failure)
- The termination of a local/remote/interactive session by the session locking mechanism
- Initiation/termination/failure of the SSH trusted channel to syslog server
- Initiation/termination/failure of the SSH trusted path with Admin
- Application of rules configured with the 'log' operation by the packet filtering function
- Indication of packets dropped due to too much network traffic by the packet filtering function

In addition the following management activities of TSF data are recorded:

- configure the access banner;
- configure the session inactivity time before session termination;
- configure the authentication failure parameters for FIA_AFL.1;
- Ability to configure audit behaviour;
- configure the cryptographic functionality;
- configure thresholds for SSH rekeying;
- re-enable an Administrator account;
- set the time which is used for time-stamps.

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 [ECG1][ECG2][ECG3] must be configured.

As a minimum, Junos OS records the following with each log entry:

- date and time of the event and/or reaction
- type of event and/or reaction
- subject identity (where applicable)
- the outcome (success or failure) of the event (where applicable).

In order to identify the key being operated on, the following details are recorded for all administrative actions relating to cryptographic keys (generating, importing, changing and deleting keys):

- CAK – imported key reference is recorded in syslog
- SAK – Key Identifier is recorded in syslog
- KEK, SAK, ICV – key references provided by process id
- SSH session keys– key reference provided by process id
- SSH keys ***generated*** for outbound trusted channel to external syslog server
- SSH keys ***imported*** for outbound trusted channel to external syslog server
- 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

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

SSH keys ***generated*** for outbound trusted channels are uniquely identified in the audit record by the public key filename and fingerprint. For example:

> Sep 27 23:36:49 yeti ssh-keygen [67873]: Generated SSH key file /root/.ssh/id_rsa.pub with fingerprint SHA256:g+7lsR7x4lQb1JT8Q3scfb2sOl8lyccojGdmkmw4dwM

SSH keys ***imported*** for use in establishing outbound trusted channels are uniquely identified in the audit record by the hash of the key imported and the username importing (to which the key will be bound).

It should be noted that 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).

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

Syslog can be configured to store the audit logs locally (***FAU_STG_EXT.1***), and optionally to send them to one or more syslog log servers in real time via Netconf over SSH. ***FMT_MOF.1/Functions***). Local audit log are stored in /var/log/ in the underlying filesystem. Only

a Security Administrator can read log files, or delete log and archive files through the CLI interface or through direct access to the filesystem having first authenticated as a Security Administrator. 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.

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 by overwriting the olders archived file.

In addition to the maximum amount of memory allocated to the archived log files being exhausted, there is also the possibility of the appliance's file system space being exhausted. File system exhaution is handled by the OS routines and not the audit functions. 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. When the filesystem reaches 92% storage capacity available on the file system, 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.

7.2 Cryptographic Support

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

7.2.1 Algorithms and Zeroization

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

- Quicksec (Inside Secure) for Junos OS 22.3R1 – JUNOS 22.3R1 QuickSec
- OpenSSL for Junos OS 22.3R1 (based on 1.1.1n) – JUNOS 22.3R1 OpenSSL
- LibMD for Junos OS 22.3R1 (the library is created from same sources as OpenSSL version, namely 1.1.1n) - JUNOS 22.3R1 LibMD
- Kernel for Junos OS 22.3R1 (based on FreeBSD-11 Stable release) - JUNOS 22.3R1 Kernel
- MACsec for Junos OS 22.3R1 – JUNOS 22.3R1 MACsec

The TOE CAVP validation certificate references for all FIPS-approved cryptographic functions implemented by the TOE are given in Table 6.

Table 6 CAVP References

Library Implemented	SFRs Supported	Function, Usage, Algorithm, Mode, Key Size	CAVP Certificate Number

Junos OS 22.3R1 MACsec	FCS_COP.1(5)	MACsec AES Data Encryption/Decryption with AES-KW, AES-GCM with key sizes 128 bit and 256 bit.	A4416
Junos OS 22.3R1 MACsec Junos OS 22.3R1 QuickSec	FCS_COP.1/ KeyedHashCMAC	MACsec AES-CMAC Keyed Hashing with key sizes 128 bit and 256 bit	A4416
	FCS_MKA_EXT.1	MACsec Key Agreement (MKA) in accordance with IEEE 802.1X-2010 and 802.1Xbx-2014 using AES-CMAC (including AES-KW and AES-CMAC)	A4416
Junos OS 22.3R1 MACsec	FCS_MACSEC_EXT.1 FCS_MACSEC_EXT.2 FCS_MACSEC_EXT.4	MACsec Data Encryption and Decryption in accordance with IEEE802.1AE-2006 using AES-GCM with key sizes 128 bit and 256 bit	AES3969 AES4544 AES4545 AES4550
Junos OS 22.3R1 Kernel Junos OS 22.3R1 OpenSSL	FCS_RBG_EXT.1	Random bit generation with HMAC-DRBG, HMAC-SHA2-256	A4417 A4210 A4419
Junos OS 22.3R1 OpenSSL	FCS_CKM.1	SSH Key Generation for RSA	A4210
	FCS_SSHS_EXT.1 FCS_COP.1/ DataEncryption FCS_COP.1/Hash FCS_COP.1/ KeyedHash FCS_COP.1/SigGen FCS_CKM.1 FCS_CKM.2	SSH AES Data Encryption/Decryption AES-CBC and AES-CTR with key sizes 128 bit and 256 bit SSH Hashing with SHA1, SHA2-256, SHA2-384, SHA2-512 SSH Keyed-hashing with HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-512 SSH Signature generation and verification using RSA with a 2048-bit and 4096-bit keys. SSH signature generation and verification using ECDSA on P-256 w/SHA-256, P-384 w/SHA-384, P-521 w/SHA-512 SSH Key generation for ECDH SSH Key generation for RSA SSH RSA Key Agreement including keypair generation. SSH EC Key Agreement including keypair generation using EC (P-256, SHA-256),ED (P-384, SHA-384), EE (P-521, SHA-512)	A4210 A4419

	FPT_TUD_EXT.1	Trusted Update signature verification using ECDSA on P-256 w/SHA-256, P-384 w/SHA-384, P-521 w/SHA-512	
Junos OS 22.3R1 LibMD	FCS_COP.1/Hash FPT_APW_EXT.1 FPT_TST_EXT.1	Cryptographic hashing for password conditioning, password hashing, and self-testing (verifying integrity of system files) using HMAC-SHA1, HMAC-SHA2-256.	A4208

All random number generation by the TOE is performed in accordance with NIST Special Publications SP 800-90A and SP 800-90B using HMAC_DRBG implemented in the OpenSSL library and kernel library. The HMAC_DRBG algorithm is seeded using an software-based entropy source implementing in accordance with NIST Special Publication SP 800-90B containing a minimum of 256 bits of entropy. ***(FCS_RBG_EXT.1.1).*** 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.**

All FIPS approved algorithms are applied when the FIPS mode is enabled[16]. The relevant FIPS knobs are specified in [ECG1][ECG2][ECG3]. (***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***)

Asymmetric keys are also generated in accordance with FIPS PUB 186-4 Appendix B.3.3 for RSA Schemes and Appendix B.4.2 for ECC Schemes for SSH communications. The TOE implements Diffie-Hellman group 14, using the modulus and generator specified by Section 3 of RFC3526. (***FCS_CKM.2, FCS_CKM.1***).

A mapping of the cryptographic algorithms to the protocols implemented by the TOE is given in Table 7. The TOE acts as both sender and recipient for MACsec and only as the server for SSH.

Table 7 Cryptographic Algorithms and Protocols

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-sah2-nistp521	AES CTR 128 AES CTR 256 AES CBC 128 AES CBC 256	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-512
MACsec	N/A	GMAC	AES GCM128 AES GCM 256	(as provided by AES GCM)
MKA	AES Key Wrap (CMAC mode)	Static-CAK (preshared)	AES-CBC 128 AES-CBC 256[17]	(as provided by AES CMAC)

The HMAC algorithms and their characteristics are stated in Table 8.

[16] 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.

Table 8 HMAC Algorithms

	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

Junos OS handles zeroization for all CSP, plaintext secret and private cryptographic keys according to Table 9. (***FCS_CKM.4***).

Table 9 Zeroization of Keys and CSP

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 the appliance is recommissioned, the config files (including CSP files such as SSH keys) are removed using the “request vmhost zeroize no-forwarding” option.
	Loaded into memory to complete session establishment	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
SSH Session Key	Session keys used with SSH, AES 128, 256, hmac-sha-1, hmac-sha2-256 or hmac-sha2-512 key (160, 256 or 512), DH Private Key (2048 or elliptic curve 256/384/521-bits)	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
User Password	Plaintext value as entered by user	Plaintext as entered	Processed in Memory	Memory free() operation is performed by Junos upon completion of authentication
		Hashed when stored (HMAC-SHA1)	Stored on disk	When the appliance is recommissioned, the config files (including the obfuscated password) are removed using the “request vmhost zeroize no-forwarding” option.
RNG State	Internal state and seed key of RNG	Plaintext	Memory	Handled by kernel, overwritten with zero’s at reboot.
MACsec CAK	Pre-shared, static Connectivity Association Key	Encrypted using AES using System Master Password	stored in config file	Actively zeroized using "request vmhost zeroize no-forwarding"

CSP	Description	Method of storage	Storage location	Zeroization Method
MACsec SAK	Security Association Key	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
MACsec KEK	Key Encryption Key	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
MACsec ICK	Integrity Check Key	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
System Master Password	Password used to derive encryption key used for protecting MACsec CAK	Plaintext	disk	Actively zeroized using "request vmhost zeroize no-forwarding"

Junos OS does not provide a CLI interface to permit the viewing of keys. Cryptographic keys are protected through the enforcement of kernel-level file access rights, limiting access to the contents of cryptographic key containers to processes with cryptographic rights or shell users with root permission[18]. (***FPT_SKP_EXT.1***)

7.2.2 SSH

Junos OS supports and enforces Trusted Channels that protect the communications between the TOE and a remote audit server from unauthorized disclosure or modification. It also supports Trusted Paths between itself and remote administrators so that the contents of administrative sessions are protected against unauthorized disclosure or modification. (***FTP_ITC.1, FTP_TRP.1/Admin***)

Junos OS provides an SSH server to support Trusted Channels using SSHv2 protocol which ensures the confidentiality and integrity of communication with the remote audit server. Export of audit information to a secure, remote server is achieved by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server. The remote audit server initiates the connection. The SSHv2 protocol ensures that the data transmitted over a SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module. (***FTP_ITC.1, FCS_SSHS_EXT.1***)

The Junos OS SSH Server also supports Trusted Paths using SSHv2 protocol which ensures the confidentiality and integrity of user sessions. The encrypted communication path between Junos OS SSH Server and a remote administrator is provided by the use of an SSH session. Remote administrators of Junos OS initiate communication to the Junos CLI through the SSH tunnel created by the SSH session. Assured identification of Junos OS is guaranteed by using public key based authentication for SSH. The SSHv2 protocol ensures that the data transmitted over a SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module. (***FTP_TRP.1/Admin, FCS_SSHS_EXT.1***)

The Junos OS SSH server is implemented in accordance with RFCs 4251, 4252, 4253, 4254, 4344, 5656 and 6668. Junos OS provides assured identification of the Junos OS appliance though public key authentication and supports password-based authentication by administrative users (Security

[18] Security Administrators do not have root permission in shell.

Administrator) for SSH connections. Conformance of the SSH implementation fo the applicable RFCs is given in Table 10.

Table 10 SSH Conformance to RFCs

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, 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	Summary	TOE implementation of Security
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 30 seconds 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 263K bytes 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	Summary	TOE implementation of Security
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 supported 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-1, SHA-256, SHA-384 and SHA-512 algorithms.**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 [NDcPP].
RFC 6668	sha2-Transport Layer Protocol	**Data Integrity Algorithms:** Both the recommended and optional algorithms hmac-sha2-256 and hmac-sha2-512 (respectively) are implemented for SSH transport.

7.2.3 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

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***).

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 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***)

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***)

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:

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

g. the MKPDU is less than 32 octets long;

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

i. 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

j. the CAK Name is not recognized.

(***FCS_MKA_EXT.1***)

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***)

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

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 11.

Table 11 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.		

Each distributed SAK shall be protected by AES Key Wrap method with Key Encryption Key (KEK) as key input (***FCS_MACSEC_EXT.4***). Each CAK is protected by AES Key Wrap (***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:

k. The SAK protected by AES Key Wrap

l. The Key Number(KN), 32 bits

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

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
- Label= "IEEE8021 SAK"
- KS-nonce = a nonce of the same size as the required SAK, obtained from an RNG each time an SAK is generated.
- MI-valuelist = a concatenation of MI values from all live participants
- KN = four octets, the Key Number assigned by the Key Server as part of the KI .
- SAKlength = two octets representing an integer value (128 for a 128 bit SAK, 256 for a 256 bit SAK) with the most significant octet first.

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***)

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***)

7.3 Identification and Authentication

Junos OS 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***)

Junos users are configured under “system login user” and are exported to the password database ‘/var/etc/master.passwd’. A Junos user is therefore 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.

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

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.

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).

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.

For password authentication, `login()` interacts with a user to request a username and password to establish and verify the user’s identity. The username entered by the administrator at the username prompt is reflected to the screen, but no feedback to screen is provided while the entry made by the administrator at the password prompt until the Enter key is pressed (***FIA_UAU.7***). `login()` uses PAM Library calls for the actual verification of this data. The password is hashed and compared to the stored value, and success/failure is indicated to `login()`, (***FIA_UIA_EXT.1***). PAM is used in the TOE support authentication management, account management, session management and password management. Login primarily uses the session management and password management functionality offered by PAM.

The retry-options can be configured to specify the action to be taken if the administrator fails to enter valid username/password credentials for password authentication when attempting to authenticate via remote access (***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.

The TOE requires users to provide unique identification and authentication data (passwords/public key) before any access to the system is granted. Prior to authentication, the only Junos OS managed responses provided to the administrator are (***FIA_UAU_EXT.2***):

- Negotiation of SSH session
- Display of the access banner
- ICMP echo responses.

Authentication data for fixed password authentication is a case-sensitive, alphanumeric value. The password has a minimum length of 10 characters and maximum length of 20 characters. Passwords may be composed of any combination of upper and lower case letters, numbers, specific special characters ("!", "@", "#", "$", "%", "^", "&", "*", "(", ")") and all other standard ASCII, extended ASCII and Unicode characters (***FIA_PMG_EXT.1***)

7.4 Security Management

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

The TOE provides user access either through the system console or remotely over the Trusted Path using the SSHv2 protocol. Users are required to provide unique identification and authentication data before any access to the system is granted. (***FMT_SMR.2, FMT_SMF.1***)

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 firmware (***FMT_MOF.1/ManualUpdate***):
  - Query currently executing version of TOE firmware (***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 (***FMT_MTD.1/CryptoKeys***):
  - SSH key generation (ecdsa, ssh-rsa)
  - Configuration of pre-shared CAKs
- 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
  - Set the system time (***FPT_STM_EXT.1***)
- Perform MACsec management functions (***FMT_SMF.1***):
  - 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

Detailed topics on the secure management of Junos OS are discussed in [ECG1][ECG2][ECG3].

7.5 Protection of the TSF

Junos OS runs the following set of self-tests during power on to check the correct operation of the Junos OS firmware (***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 firmware, 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 and iked credentials, such as Cas, CERTS, and various keys.
- Authentication error – verifies that veriexec is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real.
- Kernel, libmd, OpenSSL, QuickSec, SSH – verifies correct output from known answer tests for appropriate algorithms.

Juniper Networks devices run only binaries supplied by Juniper Networks. Within the package, each Junos OS firmware image includes fingerprints of the executables and other immutable files. Junos firmware will not execute any binary without a validating registered fingerprint. This feature protects the system against unauthorized firmware 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.

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 the [ECG1][ECG2][ECG3].

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 [ECG1][ECG2][ECG3]. (***FPT_FLS.1, FPT_TST_EXT.1,***)

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.

Security Administrators are able to query the current version of the TOE firmware using the CLI command "show version" (***FPT_TUD_EXT.1***) and, if a new version of the TOE firmware is available, initiate an update of the TOE firmware. Junos OS does not provide partial updates for the TOE, customers requiring updates must migrate to a subsequent release. Updates are downloaded and applied manually (there is no automatic updating of the Junos OS). The installable firmware 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***)

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

The fingerprint loader will only process a manifest for which it can verify the signature. Thus 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 it is 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***)

7.6 TOE Access

Junos enables Security Administrators to configure an access banner for local and remote SSH connections provided with the authentication prompt. The banner can provide warnings against unauthorized access to the secure switch as well as any other information that the Security Administrator wishes to communicate. (***FTA_TAB.1***)

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 Junos OS makes the current contents unreadable after the admin initiates the termination. No user activity can take place until the user re-identifies and authenticates.

The Security Administrator can set the TOE so that a user session is terminated after a period of inactivity. (***FTA_SSL_EXT.1, FTA_SSL.3***) For each user session Junos OS maintains a count of clock cycles (provided by the system clock) since 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.

Junos OS 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. This mechanism is the inactivity timer for administrative sessions. The Security Administrator can configure this inactivity timer on administrative sessions after which the session will be logged out.

7.7 Trusted path/Trusted Channels

The TOE supports SSH v2 for trusted channel implementation to Syslog server. The TOE supports SSH v2 (remote CLI) for secure remote administration of the TOE. Additionally, the TOE implements the MACsec protocol for the protection of layer 2 communications between itself and authenticated MACsec peers. ***(FTP_ITC.1, FTP_TRP.1)***

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