Security Target

Juniper Junos OS 23.4R1 for MX10004, MX10008 and MX10016

ST Version: 1.0

Date: December 16, 2024

Prepared for:

Juniper Networks

Prepared by:

TERON LABS

www.teronlabs.com

Revision History

Version	Date	Author(s)	Description of Change
1.0	16/12/2024	Teron Labs	Final release version

Contents

List of Tables

1 Security Target Introduction

This section is the Security Target introduction. It describes the Target of Evaluation (TOE) in a narrative way at three levels of abstraction: TOE Reference, TOE Overview and TOE Description. The three assist the reader in understanding the TOE and in determining that the TOE is suitable for the intended use.

The target audience is the users and the potential users of the TOE wishing to gain a precise understanding of the security features the TOE implements. The readers are assumed to possess a good understanding of the computer networking terms and practices. The readers are also expected to have a good understanding of network and computer security. Finally, the readers are also assumed to be proficient in the Common Criteria and the terminology thereof. Some familiarity with the networking products of Juniper Networks is beneficial.

The Security Target (ST) Introduction commences with the statements of the Security Target Reference and the TOE Reference in Sections 1.1 and 1.2, respectively. The statement of the references is followed by the TOE Overview in Sect. 1.3. The TOE Description is given in Sect. 1.4.

The TOE and the ST claim conformance to Common Criteria CCv3.1 Revision 5. The TOE claims conformance to the Protection Profile and a Protection Profile Module in accordance with a Protection Profile Configuration as identified in Table 1. The Terms given are used throughout the Security Target.

Table 1 Key Terms and References

Term	Reference
Base-PP	collaborative Protection Profile for Network Devices Version: 2.2e, Date: 23-March-2020 (cpp_nd_v2.2e)
PP-Module	PP-Module for MACsec Ethernet Encryption Version: 1.0, 2023-03-02 (mod_macsec_v1.0)
PP-Configuration	PP-Configuration for Network Devices and MACsec Ethernet Encryption Version: 1.0, 2023-03-29 (CFG_NDcPP-MACsec_V1.0)

1.1 Security Target Reference

Security Target Title	Security Target Juniper Junos OS 23.4R1 for MX10004, MX10008 and MX10016
Security Target Version	1.0
Security Target Date	December 16, 2024

1.2 TOE Reference

TOE Identification	Juniper Junos OS 23.4R1 for MX10004, MX10008 and MX10016
TOE Developer	Juniper Networks
Evaluation Sponsor	Juniper Networks

1.3 TOE Overview

1.3.1 Intended Method of Use

The TOE is a non-virtual and non-distributed network device. It is an appliance meeting the security requirements stated in the Base-PP, the PP-Module, and the Functional Package. The Base-PP and the PP-Module are used in accordance with the PP-Configuration.

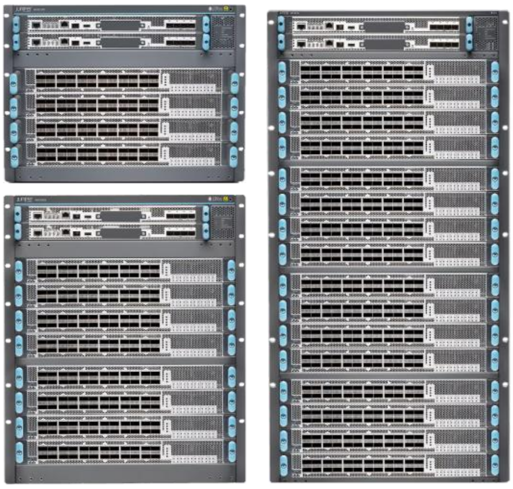

Figure 1 TOE Variants: MX10004 on the top left, MX10008 on the bottom left, and MX10016 on the right

The Juniper Networks MX10004, MX10008 and MX10016 Universal Routing Platforms are illustrated in Figure 1. They are the variants of the TOE described in this ST. They are instances of the Juniper Networks portfolio of the software-defined networking (SDN)-enabled routing platforms.

The primary function of the TOE is the secure interconnection of two networks and to regulate the traffic between the networks. The MX10004, MX10008 and MX10016 are SDN-enabled and deliver up to 76.8 Tbps of system capacity in a 4, 8 or 16 slot chassis supporting dense 100 GbE and 400 GbE interfaces. The TOE is deployed so that all traffic to and from the connected networks flows through them.

Two or more instances of TOE may be interconnected, and each interconnection may be secured at the link layer by MACsec protocol. Any of the two parties may act as the initiator of the MACsec connection. The two parties establish the symmetric MACsec key in accordance with the MACsec key agreement protocol and use the key to secure the communication. MACsec is accelerated with the hardware implementation on the dedicated line cards.

The TOE are intended for deployment at the edge to optimize converged-mobility, Internet of Things (IoT), enterprise, and cable environments. Additionally, they may be used in multiservice edge and converged core architectures. They support label-switching router (LSR), provider edge, Internet peering, and backbone applications for national or regional deployments.

The TOE are composed of the chassis, the line cards, the routing enginer and the Junos OS Operating System. In concert, they implement the routing and management plane functions for a complete network appliance. The Juniper Networks MX Series includes a wide range of appliances intended for different deployments. The MX10004, MX10008 and MX10016 are intended for deployment in data centers to provide high powered services for the users. The physical and operational characteristics of the three are summarized in Table 2.

Table 2 Characteristics of the TOE Variants

Characteristic	MX10004	MX10008	MX10016
Physical Dimensions (Width x Height x Depth)	17.4 x 12.2 x 35 in (44.2 x 33 x 88.9 cm); 42.2 in (107.7 cm) depth with electromagnetic interference (EMI) door	17.4 x 22.55 x 32 in (44.2 x 57.76 x 81.28 cm); 39.37 in (100 cm) depth with electromagnetic interference (EMI) door	17.4 x 36.6 x 35 in (44.2 x 92.96 x 88.9 cm)
Maximum Weight	272 lb (123 kg) (excluding line cards)	330 lb (150 kg) (excluding line cards)	Max. 596 lb (270.34 kg) (excluding line cards)
Mounting System	4 post rack	4 post rack	8 post rack
Typical Power Consumption	7.5kW, fully loaded	12 kW, fully loaded	14kW, fully loaded
Operating Temperature	32° to 115° F (0° to 46° C) at sea level	32° to 115° F (0° to 46° C) at sea level	32° to 115° F (0° to 46° C) at sea level

The TOE implements all security functions required for controlling access to the management functions. The management functions of the TOE are only accessible to legitimate administrators through a Command Line Interface (CLI). No other means but the CLI are available for administering the TOE. The TOE may be managed locally or remotely. Remote management sessions are protected with Secure Shell (SSH).

1.3.2 Major Security Features of the TOE

The TOE implements a set of security functions and security mechanisms required for conformance with the Base-PP and the PP-Module. The major security features implemented by the TOE are the following:

1. Security Audit. The TOE implements an audit function to collect detailed information about the state of the TOE to allow the administrator to troubleshoot the TOE and investigate possible security-related incidents.
2. Cryptography. The TOE implements a suite of cryptographic algorithms and protocols. Each cryptographic algorithm implemented by the TOE is validated against the Cryptographic Algorithm Validation Program (CAVP). The cryptographic algorithms and protocols are used to implement the critical security functions of the TOE but are also used for implementing the essential network security features:
    a. The TOE implements MACsec in accordance with IEEE 802.1AE to allow two instances of a TOE to be interconnected so that the interconnection is secured at the link layer.
    b. In addition to MACsec, the TOE implements trusted paths and trusted channels to allow remote IT systems - specifically, an audit server and the remote management station - to connect to the TOE in a secure manner. The additional trusted paths and trusted channels are implemented with SSH Protocol.
3. Identification, Authentication, Authorization and Access Control. The TOE ensures that access to the administrative functions is only granted to successfully identified and authenticated users. Illegitimate users are deterred and prevented from gaining access.
4. Security Management. The TOE implements a Command Line Interface (CLI) made available to the administrators. The CLI may be accessed locally from console or remotely over a SSH connection.
5. Protection. The TOE protects itself from tampering by passive and active means to ensure that the TOE always boots into a secure state and remains so when operated.

1.3.3 TOE Type

The TOE is a network appliance implementing the security features required for strict conformance with the Base-PP and the PP-Module. The PP and the PP-Module are used in accordance with the PP-Configuration. The TOE is neither a distributed nor a virtual network device.

1.3.4 Non-TOE Hardware, Software and Firmware

The TOE is the entire network appliance. Yet, it does require external IT devices to be properly operated. Specifically, the TOE requires the following items in the network environment:

- Syslog server including a SSHv2 client for connecting to the TOE for the TOE to send audit logs,
- A management station with a SSHv2 client for remote administration of the TOE, and
- A management station with a serial connection client for local administration of the TOE.

1.3.5 Disallowed Protocols and Services

The following protocols and services must not be used in association with the TOE:

- Telnet must not be used. It is not considered secure and violates the trusted path and trusted channel requirements.
- FTP must not be used. It is not considered secure and violates the trusted path and trusted channel requirements.
- SNMP must not be used. It is not considered secure and violates the trusted path and trusted channel requirements.
- NTP must not be used. It is not included in the certified configuration of the TOE. Only a local clock of the TOE must be used.
- SSL and TLS must not be used, including management of the TOE via J-Web, JUNOScript and JUNOScope. Neither is included in the certification and, therefore, must not be used.
- No user must be assigned super-user or Linux root account privileges. All administration of the TOE must be through the CLI.

1.4 TOE Description

1.4.1 Physical Scope of the TOE

The TOE is a network device as illustrated in Figure 2. The TOE includes the hardware, i.e., the chassis of the TOE. The Chassis implements the casing and the physical ports, the line cards, the routing engine and the hardware foundation for all those functions of the TOE which are implemented in hardware. The packet forwarding engine is part of the TOE software to perform high level packet forwarding functions on top of the hardware-based link layer and routing engine operations.

The KVM Hypervisor virtualizes the hardware for the software parts of the TOE. The software together with the line cards and the routing engine implement all routing plane and management plane functions of the TOE. The software also includes the Juniper Junos operating system.

The TOE is connected to the management console and to a syslog server. The management console may be local or remote. The TOE is also connected to the networks which it interconnects. Only the routing plane functions are implemented on the network traffic to and from the interconnected networks. All management plane functions are implemented on the devices connected to the dedicated management ports of the TOE.

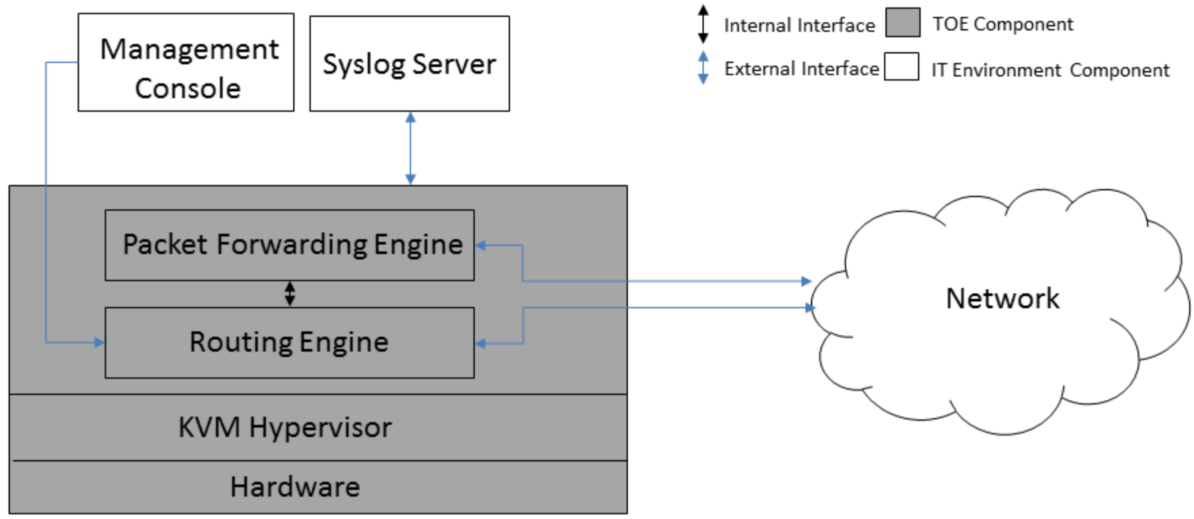

Figure 2 TOE Architecture

The TOE implements the following distinct sets of interfaces:

1. The operationally required interfaces. These include the power management and the mechanical interfaces used for the cooling and ventilation of the TOE as well as the LEDs informing the user of the status of the TOE.
2. Network interfaces used for connecting the TOE to the interconnected networks. They are the interfaces for the ingress and egress network traffic and are physically separate from all other network interfaces. The TOE implements the functionality for the network traffic to traverse through it but does not implement any security functions for processing the data on the network interfaces. An exception to this is the MACsec port filtering implemented on all MACsec traffic.
3. Management interfaces are used by the administrators to manage the TOE. Management interface is through dedicated network ports and may be accessed locally from console or remotely over a SSH connection. The management interface implements a Command Line Interface which is the only means of administering the TOE.

The physical scope of the TOE includes all hardware and software parts and the security guidance of the TOE. The parts of the TOE included in the physical scope are detailed in Table 3.

Table 3 Parts Included in the Physical Scope of the TOE

Part of the TOE	Identification	Description
Chassis	MX10004 MX10008 MX10016	The hardware platform and the casing of the TOE. Includes the 10-core 2.2GHz Intel processor with 64 or 128 GB of memory and two 200 GB solid state drives (SSD) for storage.
Line Card	LC480 or LC9600	The line cards for the MX10000 line of modular platforms are based on the Juniper Trio 6 silicon. Each slot on the MX10004, MX10008 and MX10016 supports 9.6 Tbps, while the line cards support multi-rate 1GbE, 10GbE, 25GbE, 40GbE, 50GbE,

		100GbE, and 400GbE interfaces. The LC480 line card leverages Trio 4 chips to achieve 480 Gbps total throughput. The LC9600 line card leverages Trio 6 chips to achieve 9.6 Tbps total throughput. The MX10004 and MX10008 may be fitted with either LC480 or LC9600 linecard. The MX10016 may only be fitted with LC480
Routing Engine	JNP10K-RE1	The Routing Engine, embedded in a Routing and Control Board (RCB) performs all route-processing functions. One or two RCBs may be installed on the TOE. Each RCB functions as a unit. The TOE is shipped with a single RCB installed but another one may be added for redundancy.
Junos OS	Junos OS 23.4R1	The Junos OS included in the TOE is Junos OS 23.4R1. The Junos OS includes the KVM Hypervisor. TOE software is distributed as the following Junos installation package which is identical for both variants of the TOE: junos-vmhost-install-mx-x86-64-23.4R1.9.tgz
Security Guidance	Juniper Junos OS 23.4R1 for MX10004, MX10008 and MX10016 Common Criteria Guidance Supplement v1.0	The Common Criteria Guidance supplement for the TOE. The TOE is to be always operated in accordance with the security guidance. The security guidance is distributed as a document in PDF format.

1.4.2 Logical Scope of the TOE

The TOE implements the security functionality required by the Base-PP and by the PP-Module. The major security features of the TOE are summarized in Table 4.

Table 4 Major Security Features of the TOE

Security Feature	Description
Security Audit	The TOE implements an audit function. A rich set of audit data is collected and stored as audit records. Each audit record includes a time stamp stating the exact time at which the audit record was generated. Each audit record also includes sufficient information to allow administrators of the TOE to examine the events and investigate possible security violations and attempts thereof. Audit records are stored in log files within the TOE. The administrator may also configure the TOE to forward the audit records to an external syslog server. The syslog server is not part of the TOE. Forwarding the audit records to a syslog server takes place over a trusted channel.
Cryptography	The TOE implements cryptography on hardware and software. The underlying cryptography for the trusted paths and trusted channels is implemented in software whereas the MACsec cryptography is partially implemented in hardware. Each cryptographic algorithm implemented by the TOE is CAVP-validated. This fulfills the requirements of the NIAP Policy Letter #5: Applicability and Relationship

	of NIST Cryptographic Algorithm Validation Program (CAVP) and Cryptographic Module Validation Program (CMVP) to NIAP's Common Criteria Evaluation and Validation Scheme (CCEVS).
MACsec	The TOE implements MACsec in conformance with the IEEE 802.1AE standard. The line cards of the TOE implement MACsec between adjacent devices to protect all traffic communicated between the devices. The protected traffic includes frames for Link Layer Discovery Protocol (LLDP), Dynamic Host Configuration Protocol (DHCP), Address Resolution Protocol (ARP), Spanning Tree Protocol (STP) as well as Ethernet Control frames. Destination and source Media Access Control (MAC) addresses in MACsec and MACsec Key Agreement (MKA) frames are excluded. MACsec can be deployed in point-to-point mode or shared mode with multiple stations. In the certified configuration MACsec must be configured individually on each point-to-point Ethernet link so that a pair of MACsec-capable devices (connected by a physical medium) protect Ethernet frames switched or routed from one device to the other. The two MACsec-capable devices are provided with a Connectivity Association Key (CAK) and utilize the MKA protocol to create a secure tunnel. MKA is used by the two MACsec-capable devices to agree upon MACsec keys. MACsec must be configured to protect all traffic between the devices, except for the MKA or Ethernet control traffic such as Extensible Authentication Protocol (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 Security Association (SA) and a receive SA. Once the SAs are set up, MACsec-protected frames traverse the unprotected link.
Identification, Authentication, Authorization and Access	The TOE does not implement general purpose computing facilities. Access is only granted to legitimate administrators. The TOE implements an authentication window for each attempted connection and displays a banner on that window. The administrator may configure the content of the banner to inform unauthorized users of the restricted nature of access and the consequences of attempted unauthorized access. Each user is identified with a username and authenticated with a password. Only upon successful identification and authentication is the user granted access to the TOE. The TOE implements protective measures against attempted password guessing. Each user is assigned a retry counter which keeps track of the number of consecutive failed authentication attempts on that user account. If the number exceeds the administrator-configurable number of consecutive failed authentication attempts, the account is locked for a period of time. Each user may terminate their own session and the TOE also implements an inactivity timer for each account. If the inactivity timer reaches the maximum allowed time of inactivity, the TOE terminates the session, and the user is required to re-authenticate to re-establish access.
Security Management	The TOE implements a CLI accessible to the successfully authenticated administrators. The CLI may be accessed locally from console or remotely over a SSH connection. the CLI implements the entire human user interface of the TOE. There are no alternative methods of administering the TOE. Administrators may use the CLI for performing a wide range of security management tasks on the TOE.

Protection	The TOE protects itself by passive and active means. Passive protection is achieved through the construction of the TOE. The TOE is a dedicated appliance with a restricted interface. It does not provide general computing capabilities. Access is restricted to authorized administrators and all administrator accesses are through a CLI. Administrators have no root access to the underlying Linux operating system. The network interfaces are physically separate from the management ports and may not be used for administering the TOE. The TOE implements a set of security measures for protecting the functions it implements and the configuration parameters. The TOE also maintains a clock which is used for generating time stamps and implementing various times used in the enforcement of security functions. The TOE implements self-tests at the start-up and takes protective measures in case of a failure of self-tests. Further, the TOE also allows upgrading of the software in case of vulnerabilities being discovered in the implementation.
SSH Server	The TOE implements a SSH Server. SSH is used for two key functions: It allows the administrator to access the CLI from a remote management station, and it allows a connection to the TOE from a syslog server to which audit records are forwarded. Use of SSH ensures that both remote accesses are secure. The SSH implementation of the TOE allows both password-based and public key-based authentication and implements a suite of cryptographic algorithms allowed by the Base-PP.
Trusted Paths and Channels	The TOE implements secure accesses for the administrators to manage the TOE remotely and secure protocols for connecting the TOE to external IT systems. The administrators may connect to the TOE from a remote management station using SSH. The CLI is made accessible over SSH to successfully identified and authenticated administrators. The TOE may additionally be connected from remote IT systems over SSH. SSH may be used for connecting the TOE to a syslog server.

2 Conformance Claims

This section states the Conformance Claims for the ST and the TOE. This includes a statement of the Conformance Claims, a statement of the Conformance Claim Rationale, and the Identification of the Technical Decisions applicable to the TOE.

2.1 Statement of Conformance Claims

The ST and the TOE claim conformance to Common Criteria Version 3.1 Revision 5, Part 1 through to Part 3 identified in the following:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, April 2017, Version 3.1 Revision 5, CCMB-2017-04-001
- Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, April 2017, Version 3.1 Revision 5, CCMB-2017-04-002
- Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, April 2017, Version 3.1 Revision 5, CCMB-2017-04-003

The ST claims CC Part 2 conformance as CC Part 2 Extended.

The ST claims CC Part 3 conformance as CC Part 3 conformant.

The ST claims conformance to the following Protection Profile, and the Protection Profile Module:

- collaborative Protection Profile for Network Devices Version: 2.2e, Date: 23-March-2020 (cpp_nd_v2.2e). This is the Base-PP for the evaluation and certification.
- PP-Module for MACsec Ethernet Encryption Version: 1.0, 2023-03-02 (mod_macsec_v1.0)

Conformance to the Base-PP and the PP-Module is claimed in accordance with the PP-Configuration:

- PP-Configuration for Network Devices and MACsec Ethernet Encryption Version: 1.0, 2023-03-29 (CFG_NDcPP-MACsec_V1.0)

The ST claims no conformance to any Evaluation Assurance Level or any other security assurance requirement package. Instead, the security assurance requirements applicable to the TOE are those drawn from the Base-PP as required by Sect. 2.2 of CFG_NDcPP-MACsec_V1.0.

The ST claims conformance to the collaborative Protection Profile for Network Devices Version: 2.2e, Date: 23-March-2020 (cpp_nd_2.2e) as PP-conformant.

The ST claims conformance to the PP-Configuration for Network Devices and MACsec Ethernet Encryption Version: 1.0, 2023-03-29 (CFG_NDcPP-MACsec_V1.0) as PP-configuration-conformant.

The ST claims exact conformance to the Base-PP, exact conformance to the PP-Module, and exact conformance to the PP-configuration[1].

[1] Exact conformance is defined in *CC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017)*.

2.2 Conformance Claim Rationale

2.2.1 TOE Type Consistency Rationale

The TOE is a network appliance implementing the security features required for exact conformance with the Base-PP and with the PP-Module. The PP and the PP-Module are used in conformance with the PP-Configuration. These are exactly the PP, the PP-Module, and the PP-configuration claimed in Sect. 2.1. The PP and the PP-Module are exactly as identified in Sect. 1.3 of the PP-Configuration. This ensures that the TOE Type is consistent with the TOE Type in the Base-PP, PP-Module, and PP-Configuration.

2.2.2 Security Problem Definition Consistency

The statement of the Security Problem Definition in this ST is reproduced exactly from the Base-PP and from the claimed PP-Module. The resulting Security Problem Definition is a union of the Security Problem Definition of the Base-PP and the PP-Module. There are no additional Security Problem Definition elements defined in the Functional Package. This ensures that the Security Problem Definition is consistent with the PP-Configuration.

2.2.3 Security Objective Consistency

The statement of the Security Objectives in this ST is reproduced exactly from the Base-PP and PP-Module. The resulting Security Objectives statement is a union of the Security Objectives of the Base-PP and the PP-Module. This ensures that the statement of the Security Objectives is consistent with the PP-Configuration.

2.2.4 Security Requirements Consistency

The security functional requirements are drawn exactly from the Base-PP and the PP-Module. The statement of the security functional requirements includes all mandatory and selection-based requirements. The developer claims no optional requirements. As such, the requirements are consistently drawn and ensure the consistency of the Security Functional Requirements.

The security assurance requirements are drawn from the Base-PP only. This is consistent with Sect. 2.2 of the PP-Configuration. This ensures the consistency of the Security Assurance Requirements.

2.3 Technical Decisions

The Technical Decisions (TD) applicable to the Base-PP are given in Table 5. For each TD, the applicability to the ST is stated. For each TD which is not applicable, a brief justification for the exclusion is given.

Table 5 Technical Decisions applicable to the Base-PP

TD	Description	Applicable	Exclusion Rationale (if applicable)
TD 0800	Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	No	The TOE does not claim IPSec
TD 0792	NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes	
TD 0790	NIT Technical Decision: Clarification Required for testing IPv6	No	The TOE does not claim DTLS or TLS
TD 0738	NIT Technical Decision for Link to Allowed-With List	Yes	

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

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

The Technical Decisions (TD) applicable to the PP-Module are given in Table 6. For each TD, the applicability to the ST is stated. For each TD which is not applicable, a brief justification for the exclusion is given.

Table 6 Technical Decisions Applicable to the PP-Module

TD	Description	Applicable	Exclusion Rationale (if applicable)
TD 0891	Correlation of Implicitly Satisfied Requirements when CPP_ND_V3.0E is the Base-PP	Yes	
TD 0889	Correction For Tests Incorrectly Requiring Group MACsec	Yes	
TD 0884	Expansion of Permitted EtherTypes in FCS_MACSEC_EXT.1.4	Yes	
TD 0882	MACsec Data Delay Protection, Key Agreement, and Conditional Support for Group CAK	Yes	
TD 0881	Correction to MN Usage for FPT_RPL.1 Test	Yes	
TD 0870	Security Objectives Rationale for MOD_MACSEC_V1.0	Yes	
TD 0840	Alignment of Test 22.1 to FMT_SMF.1/MACSEC	Yes	
TD 0826	Aligning MOD_MACSEC_V1.0 with CPP_ND_V3.0E	Yes	
TD 0816	Clarity for MACsec Self Test Failure Response	Yes	
TD 0803	Clarification for Configurable MACsec CKN Length	Yes	
TD 0746	Correction to FPT_RPL.1 Test 25	Yes	
TD 0728	Corrections to MACSec PP-Module SD	Yes	

3 Security Problem Definition

The Security Problem Definition includes a statement of the Threats, Assumptions and OSPs applicable to the TOE. Each is stated in this section.

3.1 Threats

The threats applicable to the TOE are drawn from the Base-PP and of the PP-Module. There are no additions or omissions, and the wording of each threat statement is taken verbatim from the Base-PP and the PP-Module. The threats drawn from the Base-PP as applicable to a non-distributed and non-virtual network device are given in Table 7. The threats drawn from the PP-Module are given in Table 8.

Table 7 Threats drawn from the Base-PP

Threat ID	Threat Statement
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.

	Nonvalidated 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.

Table 8 Threats drawn from the PP-Module

Threat ID	Threat Statement
T.DATA_INTEGRITY	An attacker may modify data transmitted over the layer 2 link in a way that is not detected by the recipient. Devices on a network may be exposed to attacks that attempt to corrupt or modify data in transit without authorization. If malicious devices are able to modify and replay data that is transmitted over a layer 2 link, then the data contained within the communications may be susceptible to a loss of integrity.
T.NETWORK_ACCESS	An attacker may send traffic through the TOE that enables them to access devices in the TOE's operational environment without authorization. A MACsec device may sit on the periphery of a network, which means that it may have an externally- facing interface to a public network. Devices located in the public network may attempt to exercise services located on the internal network that are intended to be accessed only from within the internal network or externally accessible only from specifically authorized devices. If the MACsec device allows unauthorized external devices access to the internal network, these devices on the internal network may be subject to compromise. Similarly, if two MACsec devices are deployed to facilitate end-to-end encryption of traffic that is contained within a single network, an attacker could use an insecure MACsec device as a method to access devices on a specific segment of that network such as an individual LAN.

T.UNTRUSTED_MACSEC_ COMMUNICATION_CHANNELS	An attacker may acquire sensitive TOE or user data that is transmitted to or from the TOE because an untrusted communication channel causes a disclosure of data in transit. A generic network device may be threatened by the use of insecure communications channels to transmit sensitive data. The attack surface of a MACsec device also includes the MACsec trusted channels. Inability to secure communications channels, or failure to do so correctly, would expose user data that is assumed to be secure to the threat of unauthorized disclosure.

3.2 Assumptions

The assumptions applicable to the TOE are drawn from the Base-PP. There are no additions or omissions, and the wording of each assumption statement is taken verbatim from the Base-PP. The assumptions drawn from the Base-PP as applicable to a non-distributed and non-virtual network device are given in Table 9. There are no additional assumptions stated in the PP-Module.

Table 9 Assumptions Drawn from the Base-PP

Assumption ID	Assumption Statement
A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.
A.LIMITED_ FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality). If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform.
A.NO_THRU_TRAFFIC_ PROTECTION	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).

A.TRUSTED_ ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_ SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

3.3 Organizational Security Policies

The Organizational Security Policies (OSP) applicable to the TOE are drawn from the Base-PP. There are no additions or omissions, and the wording of each OSP statement is taken verbatim from the Base-PP. There are no additional OSPs defined in the PP-Module.

Table 10 OSPs Drawn From the Base-PP

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

4 Security Objectives

The security objectives are stated for the TOE Sect. 4.1 and for the operational environment of the TOE in Sect. 4.2. The security objectives rationale is given in Sect. 4.3.

4.1 Security Objectives for the TOE

The security objectives for the TOE are drawn from the PP-Module. There are no security objectives for the TOE stated on the Base-PP. The security objectives for the TOE are drawn in verbatim from the PP-Module and are stated in Table 11.

Table 11 Security Objectives for the TOE Drawn from the PP-Module

Security Objective ID	Security Objective Statement
O.AUTHENTICATION_ MACSEC	To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (MKA) will allow a MACsec peer to establish connectivity associations (CAs) with another MACsec peer. MACsec endpoints authenticate each other to ensure they are communicating with an authorized MAC Security Entity (SecY) entity.
O.AUTHORIZED_ ADMINISTRATION	All network devices are expected to provide services that allow the security functionality of the device to be managed. The MACsec device, as a specific type of network device, has a refined set of management functions to address its specialized behavior. In order to further mitigate the threat of a compromise of its security functionality, the MACsec device prescribes the ability to limit brute-force authentication attempts by enforcing lockout of accounts that experience excessive failures and by limiting access to security-relevant data that administrators do not need to view.
O.CRYPTOGRAPHIC_ FUNCTIONS_MACSEC	To address the issues associated with unauthorized modification and disclosure of information, compliant TOEs will implement cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.
O.PORT_FILTERING_ MACSEC	To further address the issues associated with unauthorized network access, a compliant TOE's port filtering capability will restrict the flow of network traffic through the TOE based on layer 2 frame characteristics and whether or not the traffic represents valid MACsec frames and MACsec Key Agreement Protocol Data Units (MKPDUs).
O.REPLAY_DETECTION	A MACsec device is expected to help mitigate the threat of MACsec data integrity violations by providing a mechanism to detect and discard replayed traffic for MPDUs.
O.SYSTEM_MONITORING_ MACSEC	To address the issues of administrators being able to monitor the operations of the MACsec device, compliant TOEs will implement the ability to log the flow of Ethernet traffic. Specifically, the TOE will provide the means for administrators to configure rules to 'log' when Ethernet traffic grants or restricts access. As a result, the 'log' will result in informative event logs whenever a match occurs. In addition, the

	establishment of security CAs is auditable, not only between MACsec devices, but also with MAC Security Key Agreement Entities (KaYs).
O.TSF_INTEGRITY	To mitigate the security risk that the MACsec device may fail during startup, it is required to fail-secure if any self-test failures occur during startup. This ensures that the device will only operate when it is in a known state.

4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are drawn from the Base-PP. The PP-Module does not state security objectives for the operational environment. The security objectives for the operational environment as applicable to a non-virtual and non-distributed network device are drawn in verbatim from the Base-PP and are stated in Table 12.

Table 12 Security Objective for the Operational Environment Drawn from the Base-PP

Security Objective ID	Security Objective Statement
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_ PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.
OE.NO_THRU_TRAFFIC_ PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_ SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.RESDUAL_ INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

4.3 Security Objectives Rationale

The security objectives rationales are drawn in verbatim from the Base-PP and the PP-Module. Therefore, the security objectives rationales given in the Base-PP and in the PP-Module are directly applicable to the ST. They are not repeated here.

5 Security Requirements

This section states the security requirements applicable to the TOE. The statement commences with the extended components definition in Sect. 5.1. The statement of the extended components is followed by the statement of the notations and conventions used in the expression of the security requirements. The security functional requirements are summarized in Sect. 5.3 and stated in the subsequent subsections on a per functional class basis. The security assurance requirements are only drawn from the Base-PP and are given in Sect. 5.11. The security requirements rationale is given in Sect. 5.12

5.1 Extended Components Definition

The ST references several extended components. Each one is taken verbatim from the Base-PP or the PP-Module. Only the operations allowed in the statement of the extended components are implemented in the ST. There are no additional or modified extended components included in the ST. Therefore, the statement of the extended components is exactly as in the Base-PP and the PP-Module. They are not repeated here.

5.2 Notation and Conventions

This ST follows the specific conventions in the completion of the operations on the Security Functional Requirements. The following conventions are followed to indicate the operations:

- Unaltered Security Functional Requirements are stated using the notation given in CC Part 2 or in the applicable extended component definition.
- When a refinement made in the ST, the added text is indicated with a **bold font** and any removal of text is indicated with a ~~strikethrough~~.
- When a selection is completed in the ST, the selected values are indicated with underlined text.
    - For example, a selection "[selection: disclosure, modification, loss of use]" in a Security Functional Requirement drawn from the Base-PP or PP-Module might become "[<u>disclosure</u>]" when the selection is performed in the ST.
- Assignment completed in the ST is indicated with *italicized font*.
- Assignment completed within a selection in the ST is indicated with <u>*italicized and underlined font*</u>.
    - For example, an assignment within a selection "[selection: change_default, query, modify, delete, [assignment: other operations]]" in a Security Functional Requirement drawn from the Base-PP or PP-Module might become "[<u>change_default</u>,[<u>*select_tag*</u>]]" when both the selection and the assignment are completed in the ST.
- Iteration is indicated by adding a descriptive string starting with "/" (e.g. "FCS_COP.1/Hash").
- Extended requirements are indicated using the notation given in the Base-PP or PP-Module from which they are drawn. Each extended Security Functional Requirement is indicated with a label "_EXT" in the end of the requirement name (e.g. FCS_RBG_EXT).

When the Base-PP or the PP-Module uses an alternative notation or expression for the statement of a Security Functional Requirements, that notation or expression is followed in the ST - possibly with the addition of the above conventions. This includes, for example,

- The capitalization of the component names is followed in verbatim even if sometimes inconsistent, and
- The PP-Module alternatives for selection operations are given in italic font. The italic font is maintained and additionally also underlined to indicate that the selection is performed from the set of allowed values.

The Security Assurance Requirements are drawn from the Base-PP only for conformance with the PP-Configuration. There are no operations defined for the Security Assurance Requirements. The notation for expressing the Security Assurance Requirements is taken verbatim from the Base-PP.

5.3 Security Functional Requirements Summary

The Security Functional Requirements applicable to the TOE as drawn from different sources are summarized in Table 13.

Table 13 SFR Summary

Security Functional Class	Security Functional Components Drawn from the Base-PP
FAU: Security Audit	FAU_GEN.1 Audit Data Generation FAU_GEN.2 User identity association FAU_STG_EXT.1 Protected Audit Event Storage
FCS: Cryptographic Support	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_CKM.4 Cryptographic Key Destruction FCS_COP.1/DataEncryption Cryptographic operation (AES Data Encryption/Decryption) FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation FCS_SSHS_EXT.1 SSH Server Protocol
FIA: Identification and Authentication	FIA_AFL.1 Authentication Failure Management FIA_PMG_EXT.1 Password Management FIA_UAU.7 Protected Authentication Feedback FIA_UAU_EXT.2 Password-based Authentication Mechanism FIA_UIA_EXT.1 User Identification and Authentication
FMT: Security Management	FMT_MOF.1/Functions Management of security functions behaviour FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour FMT_MOF.1/Services Management of security functions behaviour FMT_MTD.1/CoreData Management of TSF Data FMT_MTD.1/CryptoKeys Management of TSF data FMT_SMF.1 Specification of Management Functions FMT_SMR.2 Restrictions on Security Roles

FPT: Protection of the TSF	FPT_APW_EXT.1 Protection of Administrator Passwords FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys) FPT_STM_EXT.1 Reliable Time Stamps FPT_TST_EXT.1 TSF Testing FPT_TUD_EXT.1 Trusted Update
FTA: TOE Access	FTA_SSL.3 TSF-initiated Termination FTA_SSL.4 User-initiated Termination FTA_SSL_EXT.1 TSF-initiated Session Locking FTA_TAB.1 Default TOE Access Banners
FTP: Trusted Path/Channels	FTP_ITC.1 Inter-TSF Trusted Channel FTP_TRP.1/Admin Trusted Path
Security Functional Class	**Security Functional Components Drawn from the PP-Module**
FAU: Security Audit	FAU_GEN.1/MACSEC Audit Data Generation (MACsec)
FCS: Cryptographic Support	FCS_COP.1/CMAC Cryptographic Operation (AES-CMAC Keyed Hash Algorithm) FCS_COP.1/MACSEC Cryptographic Operation (MACsec AES Data Encryption and Decryption) FCS_MACSEC_EXT.1 MACsec FCS_MACSEC_EXT.2 MACsec Integrity and Confidentiality FCS_MACSEC_EXT.3 MACsec Randomness FCS_MACSEC_EXT.4 MACsec Key Usage FCS_MKA_EXT.1 MACsec Key Agreement
FIA: Identification and Authentication	FIA_PSK_EXT.1 Pre-Shared Key Composition
FMT: Security Management	FMT_SMF.1/MACSEC Specification of Management Functions (MACsec)
FPT: Protection of the TSF	FPT_CAK_EXT.1 Protection of CAK Data FPT_FLS.1 Failure with Preservation of Secure State FPT_RPL.1 Replay Detection
FTP: Trusted Path/Channels	FTP_ITC.1/MACSEC Inter-TSF Trusted Channel (MACsec Communications)

5.4 Security Audit (FAU)

5.4.1 Security Audit Data Generation (FAU_GEN)

5.4.1.1 FAU_GEN.1 Audit data generation (Refinement)

FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;
b) All auditable events for the <u>not specified</u> 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).*
   - *<u>[no other actions]</u>].*

d) *Specifically defined auditable events listed in* Table 14.

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 cPP/ST, *information specified in column three of* Table 14.

Table 14 Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure

FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., an IP address).
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Services	None.	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (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.	Identification of the initiator and target of failed

	• Termination of the trusted channel. • Failure of the trusted channel functions.	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.

5.4.1.2 FAU_GEN.1/MACSEC Audit Data Generation (MACsec)

FAU_GEN.1/MACSEC Audit Data Generation (MACsec)

FAU_GEN.1.1/MACSEC The TSF shall be able to generate an audit record of the following auditable events:

a. Start-up and shutdown of the audit functions;

b. All auditable events for the [*not specified*] level of audit;

c. All administrative actions;

d. [Specifically defined auditable events listed in the Auditable Events table (*Table 15*)]

Table 15 Auditable Events for MACsec

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

FAU_GEN.1.2/MACSEC The TSF shall record within each audit record at least the following information:

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

b. For each audit event type, based on the auditable event definitions of the functional components included in the PP-**Module**/ST, [*information specified in column three of the Auditable Events table (*Table 15*)*].

5.4.1.3 FAU_GEN.2 User identity association

FAU_GEN.2 User identity association

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

5.4.2 Security audit event storage (Extended - FAU_STG_EXT)

5.4.2.1 FAU_STG_EXT.12 Protected Audit Event Storage

FAU_STG_EXT.1 Protected Audit Event Storage

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

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

- *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.

5.5 Cryptographic Support (FCS)

5.5.1 Cryptographic Key Management (FCS_CKM)

5.5.1.1 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*]~~.

5.5.1.2 FCS_CKM.2 Cryptographic Key Establishment (Refinement)

FCS_CKM.2 Cryptographic Key Establishment

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

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

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

[2] As per TD 0581

[3] As per TD 0580

5.5.1.3 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*.

5.5.2 Cryptographic Operation (FCS_COP)

5.5.2.1 FCS_COP.1 Cryptographic Operation

FCS_COP.1/DataEncryption Cryptographic operation (AES Data Encryption/Decryption)

FCS_COP.1.1/DataEncryption The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in [CBC, CTR] mode* and cryptographic key sizes *[128 bits, 256 bits]* that meet the following: *AES as specified in ISO 18033-3*, *[CBC as specified in ISO 10116, CTR as specified in ISO 10116]*.

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

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

- *RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits, 4096 bits],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits, 384 bits, 521 bits]*

]

that meet the following: *[*

- *For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 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/CMAC Cryptographic Operation (AES-CMAC Keyed Hash Algorithm)

FCS_COP.1.1/CMAC The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm [*AES-CMAC*] and cryptographic key sizes **[*128, 256*] bits and message digest size of 128** bits that meets the following: [*NIST SP 800-38B*].

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

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

5.5.2.2 FCS_MACSEC_EXT.1 MACsec

FCS_MACSEC_EXT.1 MACsec

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

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

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

FCS_MACSEC_EXT.1.4[4] The TSF shall permit only EAPOL (Port Access Entity (PAE) EtherType 88-8E), MACsec frames (EtherType 88-E5), and [*<u>MAC control frames (EtherType is 88-08)</u>*] and shall discard others.

5.5.2.3 FCS_MACSEC_EXT.2 MACsec Integrity and Confidentiality

FCS_MACSEC_EXT.2 MACsec Integrity and Confidentiality

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

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

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

5.5.2.4 FCS_MACSEC_EXT.3 MACsec Randomness

FCS_MACSEC_EXT.3 MACsec Randomness

FCS_MACSEC_EXT.3.1 The TSF shall generate unique Secure Association Keys (SAKs) using [*<u>key derivation from Connectivity Association Key (CAK) per section 9.8.1 of IEEE 802.1X-2010</u>*] such that the likelihood of a repeating SAK is no less than 1 in 2 to the power of the size of the generated key.

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

---

[4] As per TD0884

5.5.2.5 FCS_MACSEC_EXT.4 MACsec Key Usage

FCS_MACSEC_EXT.4 MACsec Key Usage

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

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

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

FCS_MACSEC_EXT.4.4 The TSF shall associate Connectivity Association Key Names (CKNs) with SAKs that are defined by the KDF using the CAK as input data (per IEEE 802.1X-2010, Section 9.8.1).

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

5.5.2.6 FCS_MKA_EXT.1 MACsec Key Agreement

FCS_MKA_EXT.1 MACsec Key Agreement

FCS_MKA_EXT.1.1 The TSF shall implement Key Agreement Protocol (MKA) in accordance with IEEE 802.1X-2010 and 802.1Xbx-2014.

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

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

FCS_MKA_EXT.1.4[5] The TSF shall enforce an MKA Lifetime Timeout limit of 6.0 seconds and [*MKA Bounded Hello Timeout limit of 0.5 seconds*].

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

- *pairwise CAKs that are PSKs*

].

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

FCS_MKA_EXT.1.7 The TSF shall validate MKPDUs according to IEEE 802.1X-2010 Section 11.11.2. In particular, the TSF shall discard without further processing any MKPDUs to which any of the following conditions apply:

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

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

[5] As per TD0882

a. If the Algorithm Agility parameter identifies an algorithm that has been implemented by the receiver, the ICV shall be verified as specified in IEEE 802.1X-2010 Section 9.4.1.

b. If the Algorithm Agility parameter is unrecognized or not implemented by the receiver, its value can be recorded for diagnosis but the received MKPDU shall be discarded without further processing.

Each received MKPDU that is validated as specified in this clause and verified as specified in IEEE 802.1X-2010 Section 9.4.1 shall be decoded as specified in IEEE 802.1X-2010 Section 11.11.4.

5.5.3 Random Bit Generation (Extended - FCS_RBG_EXT)

5.5.3.1 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.

5.5.4 Cryptographic Protocols (Extended)

5.5.4.1 FCS_SSHC_EXT & FCS_SSHS_EXT SSH Protocol

FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TOE shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [*4344, 5656, 6668, 8308 section 3.1, 8332,*].

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, [*password-based*][6].

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*263,000*] 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, aes128-ctr, aes256-cbc, aes256-ctr*].

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

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

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 protocols.

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.

[6] As per TD 0631.

5.6 Identification and Authentication (FIA)

5.6.1 Authentication Failure Management (FIA_AFL)

5.6.1.1 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 [unlocking of the account from console] is taken by an Administrator; prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].*

5.6.2 Password Management (Extended – FIA_PMG_EXT)

5.6.2.1 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: *["!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [all other standard ASCII, extended ASCII and Unicode characters*]];
b) Minimum password length shall be configurable to between [*10*] and [*20*] characters.

5.6.3 Pre-Shared Key Composition (FIA_PSK_EXT.1)

FIA_PSK_EXT.1 Pre-Shared Key Composition

FIA_PSK_EXT.1.1 The TSF shall use PSKs for MKA as defined by IEEE 802.1X-2010, [*no other protocols*].

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

5.6.4 Protected Authentication Feedback (FIA_UAU)

5.6.4.1 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7 Protected Authentication Feedback (Refinement)

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

5.6.5 User Identification and Authentication (Extended - FIA_UIA_EXT)

5.6.5.1 User authentication (FIA_UAU) (Extended – FIA_UAU_EXT)

6.5.4.1 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.

5.6.5.2 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.

5.7 Security Management (FMT)

5.7.1 Management of functions in TSF (FMT_MOF)

5.7.1.1 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*.

5.7.1.2 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*.

5.7.1.3 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 **start and stop** ~~the functions~~ **services** to *Security Administrators*.

5.7.2 Management of TSF Data (FMT_MTD)

5.7.2.1 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.

5.7.2.2 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*.

5.7.3 Specification of Management Functions (FMT_SMF)

5.7.3.1 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1 Specification of Management Functions

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

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

[
  - *Ability to start and stop services;*
  - *Ability to configure audit behaviour (e.g. changes to storage location for audit; changes to behaviour when local audit storage space is full);*
  - *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
  - *Ability to manage cryptographic keys;*
  - *Ability to configure the cryptographic functionality;*
  - *Ability to configure thresholds for SSH rekeying;*
  - *Ability to re-enable an Administrator account;*
  - *Ability to set the time which is used for time-stamps;*
  - *Ability to manage the trusted public key databases;[7]].*

FMT_SMF.1/MACSEC Specification of Management Functions (MACsec)

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

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

[

- *No other MACsec management functions*

]].

[7] As per TD 0631

5.7.4 Security Management Roles (FMT_SMR)

5.7.4.1 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.

5.8 Protection of the TSF (FPT)

5.8.1 Protection of Administrator Passwords (Extended – FPT_APW_EXT)

5.8.1.1 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.

5.8.2 Protection of CAK Data (FPT_CAK_EXT.1)

5.8.2.1 FPT_CAK_EXT.1 Protection of CAK Data

FPT_CAK_EXT.1 Protection of CAK Data

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

5.8.3 Failure with Preservation of Secure State (FPT_FLS.1)

5.8.3.1 FPT_FLS.1 Failure with Preservation of Secure State

FPT_FLS.1 Failure with Preservation of Secure State

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

5.8.4 Replay Detection (FPT_RPL.1)

5.8.4.1 FPT_RPL.1 Replay Detection

FPT_RPL.1 Replay Detection

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

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

5.8.5 Protection of the TSF Data (Extended - FPT_SKP_EXT)

5.8.5.1 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.

5.8.6 Time stamps (Extended - FPT_STM_EXT)

5.8.6.1 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*].

5.8.7 TSF Testing (Extended - FPT_TST_EXT)

5.8.7.1 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 [d*uring initial start-up (on power on)*] to demonstrate the correct operation of the TSF: [*Power on test, File integrity test, Crypto integrity test, Authentication test, Algorithm known answer tests*].

5.8.8 Trusted Update (FPT_TUD_EXT)

5.8.8.1 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.

5.9 TOE Access (FTA)

5.9.1 Sesssion Locking and Termination (FTA_SSL)

5.9.1.1 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*.

5.9.1.2 FTA_SSL.4 User-initiated Termination (Refinement)

FTA_SSL.4 User-initiated Termination

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

5.9.2 TSF-initiated Session Locking (Extended – FTA_SSL_EXT)

5.9.2.1 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.

5.9.3 TOE Access Banners (FTA_TAB)

5.9.3.1 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.

5.10 Trusted Path/Channels (FTP)

5.10.1 Trusted Channel (FTP_ITC)

5.10.1.1 FTP_ITC.1 Inter-TSF Trusted Channel (Refinement)

FTP_ITC.1 Inter-TSF Trusted Channel

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

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [*Forwarding of audit records to an external audit server*].

FTP_ITC.1/MACSEC Inter-TSF Trusted Channel (MACsec Communications)

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

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

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

5.10.2 Trusted Path (FTP_TRP)

5.10.2.1 FTP_TRP.1/Admin Trusted Path (Refinement)

FTP_TRP.1/Admin Trusted Path

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

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

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

5.11 Security Assurance Requirements

This section states the Security Assurance Requirements. For conformance with the PP-Configuration, the Security Assurance Requirements are drawn from the Base-PP only. The applicable Security Assurance Requirements are stated in Table 16.

Table 16 Security Assurance Requirements

Security Assurance Class	Security Assurance Components
Security Target (ASE)	Conformance claims (ASE_CCL.1) Extended components definition (ASE_ECD.1) ST Introduction (ASE_INT.1) Security objectives for the operational environment (ASE_OBJ.1) Stated security requirements (ASE_REQ.1) Security Problem Definition (ASE_SPD.1)
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 (ALS)	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)

5.12 Security Requirements Rationale

The Security Functional Requirements are drawn from the Base-PP and PP-Module and not from any other source. The ST claims exact conformance to the Base-PP and to the PP-Module. The Security Functional Requirements include each mandatory requirement and each applicable optional and selection-based requirement. Only the operations allowed in the Base-PP and the PP-Module are implemented. Therefore, the Security Functional Rationales of the Base-PP and the PP-Module are directly applicable to the ST as well. They are not repeated here.

The Security Assurance Requirements are drawn from the Base-PP only as required by the PP-Configuration. None. are added or removed. Therefore, the Security Assurance Requirements Rationale of the Base-PP is directly applicable to the ST as well. It is not repeated here.

6 TOE Summary Specification

The TOE Summary Specification includes the description how the TOE fulfills the security functional requirements, and how the developer and the evaluator fulfill the security assurance requirements. Each is described in this section. Additional details on the cryptographic algorithms and protocols implemented in the TOE are also given.

6.1 Fulfillment of the Security Functional Requirements

The fulfilment of the Security Functional Components by the TOE is given in Table 17. Each Security Functional Component applicable to the TOE is listed and the fulfilment of that component described.

Table 17 Fulfilment of the Security Functional Components

Security Functional Component	Fulfilment
FAU_GEN.1 Audit Data Generation	The TOE generates and stores audit records for several events. The list of audit events per Security Functional Component drawn from the Base-PP is given in Table 14. Auditing is implemented using syslog. 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. The audit knobs detailed in the security guidance must be configured. In the minimum, the TOE records the following information with each log entry: • Date and time of the event and/or reaction, • Type of event and/or reaction, • Subject identity (where applicable), and • The outcome (success or failure) of the event (if applicable). SSH keys used for trusted channels are not deleted by the management daemon when SSH is de-configured. SSH keys used for trusted channels are only deleted is when a `request vmhost zeroize` command is issued by the administrator. That commences zeroization of the entire appliance of which it is not possible to store an audit record.
FAU_GEN.1/MACSEC Audit Data Generation (MACsec)	In addition to the audit events required by the Base-PP, the TOE implements comprehensive audit event collection for MACsec. The additional MACsec specific audit events are listed in Table 15. MACsec-related audit event collection is also implemented using syslog. The basic data collected on MACsec-related events is identical to that collected on Base-PP related events. Additionally, the data specified in Table 15 is stored in the log entries on each MACsec-related SFR.
FAU_GEN.2 User identity association	The subject identity is the username of the human user of the TOE or the IP Address of the peer entity attempting to connect to the TOE. Cryptographic keys are identified by the following detail when generated, imported, changed, or deleted: • SSH session keys: Key reference provided by process id, including the following keys:

	o SSH keys generated for outbound trusted channel to external syslog server.
	o 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 used for authentication.
	
	For SSH (ephemeral) session keys the PID is used as the key reference to relate the key generation and key destruction. 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 such as 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 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 identified in the audit record by the hash of the key imported and the username of the user importing the key. The key is bound to the username.
	
	Cryptographic keys used for MACsec are identified by the following detail when generated, imported, changed, or deleted:
	
	• MACsec CAK: Imported key reference is recorded in the log entries.
	• MACsec SAK: Key Identifier is recorded in the log entries.
	• MACsec KEK, SAK, ICV: Key references provided by process id.

FAU_STG_EXT.1 Protected Audit Event Storage	Syslog included in the TOE can be configured to store the audit logs locally or to send them to one or more syslog log servers. Log entries are sent in real time via Netconf over SSH. Local audit logs are stored in `/var/log/` in the filesystem of the TOE. Only a Security Administrator can read, delete, or archive log files. Managing the log files is through the CLI interface or through direct access to the filesystem. The log files are automatically deleted locally if the administrator-configurable limit on the storage volume is reached. The default maximum storage size is 1Gb but the administrator can modify the allocated storage size using the `size` argument on the `set system syslog` CLI command. The TOE uses an active log file supported by a number archive files. The default number of archive files is 10 but the administrator may configure the number to be between 1 and 1000.

	When the active log file reaches the maximum size, the TOE closes the file, compresses it, and names the compressed archive file 'logfile.0.gz'. The TOE then opens and writes to a new active log file. When the new active log file reaches the maximum size, 'logfile.0.gz' is renamed 'logfile.1.gz', and the active log file is closed, compressed, and renamed 'logfile.0.gz'. This is repeated so that the latest compressed logfile is always named 'logfile.0.gz'. When the maximum number of archive files is reached or 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. A 1Gb syslog file takes approximately 0.25Gb of storage when archived. Syslog files total size may reach the complete storage capacity allocated to the `/var` filesystem. The complete storage capacity is platform specific. When the filesystem size reaches 92% of the storage capacity, an event is generated but the event daemon 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, the `/var` filesystem becomes exhausted. If the file system is exhausted, a final log entry "No space left on device" is generated and the logging is terminated. Other functions of the TOE shall continue when the audit log storage space is exhausted.
FCS_CKM.1 Cryptographic Key Generation	The TOE generates cryptographic keys using RSA Schemes, ECC Schemes, and FFC Schemes: 1. RSA schemes are used to generate cryptographic key sizes of 2048-bit or greater. The keys are generated in accordance with FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3. 2. ECC schemes are used to generate cryptographic keys for use with NIST curves P-256, P-384, and P-521. The ECC Schemes are used in accordance with FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4. 3. FFC Schemes use 'safe-prime' groups are used for generating SSH session keys. The FFC Schemes are used in accordance with NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and RFC 3526.
FCS_CKM.2 Cryptographic Key Establishment	The TOE implements key establishment using Elliptic curve-based key establishment schemes and FFC-based key establishment schemes. 1. Elliptic curve-based key establishment schemes are used in accordance with NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography". 2. FFC Schemes using "safe-prime" groups are used in accordance with 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and using groups listed in RFC 3526. The key establishment, authentication, encryption and data integrity algorithms and methods used by the TOE are detailed in Sect. 6.3.3.

<table>
<tr><td>FCS_CKM.4 Cryptographic Key Destruction</td><td>The TOE implements functions for secure erasure of cryptographic keys and Critical Security Parameters (CSK). The keys and CSPs stored in the volatile memory are typically erased by the TOE software calling the <code>free()</code> function at the termination of a session. Keys and Critical security parameters stored in the non-volatile memory are erased when the administrator decommissions the TOE.

The details of the erasure of cryptographic keys and CSPs is given in Sect. 6.3.2.</td></tr>
<tr><td>FCS_COP.1/DataEncryption Cryptographic operation (AES Data Encryption/Decryption)</td><td>The TOE implements the AES key sizes and modes of operation used for symmetric encryption and decryption as detailed in Sect. 6.3.4.</td></tr>
<tr><td>FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)</td><td>The TOE implements asymmetric cryptography for digital signature generation and verification functions and key sizes as detailed in Sect. 6.3.4.</td></tr>
<tr><td>FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)</td><td>The TOE implements cryptographic hash functions SHA-1, SHA-256, SHA-384 and SHA-512. The hash functions are used by the TOE as summarized in the following:
<table>
<tr><th></th><th>SHA-1</th><th>SHA-256</th><th>SHA-384</th><th>SHA-512</th></tr>
<tr><td>SSH Hashing</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>SSH HMAC</td><td>X</td><td>X</td><td></td><td>X</td></tr>
<tr><td>SSH RSA Key Agreement</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>SSH ECC Key Agreement</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>RSA Signature generation and verification</td><td></td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>ECDSA Signature generation and verification</td><td></td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Password hashing</td><td></td><td>X</td><td></td><td>X</td></tr>
<tr><td>File system integrity self-tests</td><td>X</td><td>X</td><td></td><td></td></tr>
<tr><td>Firmware integrity self-test</td><td>X</td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)</td><td>The TOE implements HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-512. The parameter sizes used by the difference keyed hash algorithms are the following:
<table>
<tr><th></th><th>HMAC-SHA-1</th><th>HMAC-SHA-256</th><th>HMAC-SHA-512</th></tr>
<tr><td>Key length</td><td>160 bits</td><td>256 bits</td><td>512 bits</td></tr>
<tr><td>Hash function</td><td>SHA-1</td><td>SHA-256</td><td>SHA-512</td></tr>
<tr><td>Block size</td><td>512 bits</td><td>512 bits</td><td>1024 bits</td></tr>
<tr><td>Output size</td><td>160 bits</td><td>256 bits</td><td>512b its</td></tr>
</table></td></tr>
<tr><td>FCS_COP.1/CMAC Cryptographic Operation</td><td>The TOE implements CMAC using AES-CMAC. Cryptographic key sizes of 128 bits and 256 bits. The message digest size is 128 bits for a 128 bit AES key, and</td></tr>
</table>

(AES-CMAC Keyed Hash Algorithm)	256 bits for a 256 bit key. The output used as an input is 128 bits for a 128 bit key, and 256 bits for a 256 bit key.
FCS_COP.1/MACSEC Cryptographic Operation (MACsec AES Data Encryption and Decryption)	The TOE implements MACsec encryption and decryption using AES used in AES Key Wrap and in the GCM modes of operation. 128-bit and 256-bit keys are used. AES is implemented in accordance with ISO 18033-3, AES Key Wrap is implemented in accordance with NIST SP 800-38F, and the GCM mode of operation is implemented in accordance with ISO 19772.
FCS_MACSEC_EXT.1 MACsec	The TOE implements MACsec in accordance with IEEE 802.1AE-2018. The implementation supports the following: 1. AES-128 and AES-256 ciphersuites without Extended Packet Numbering (XPN), 2. MACsec Key Agreement (MKA) protocol in a Static-CAK mode using a pre-shared key, 3. One Connectivity-Association (CA) per physical port (Interface Device, IFD), 4. One Tx-Secure Channel per CA for transmission, 5. One Rx- Secure Channel per CA for receipt, 6. 4 Secure Associations (SA) per SC The Line Card of the TOE can be programed to bypass certain EtherTypes. In the evaluated configuration, only the Extended Authentication Protocol over LAN (EAPOL - PAE EtherType 88-8E), MACsec frames (EtherType 88-E5), and MACsec control frames (EtherType is 88-08) are bypassed. This means that only these Ethernet frames will be accepted by the TO. All other frames are rejected. Also, a filter in the Packet Forwarding Engine (PFE) traps the packets to the Routing Engine (RE) with ether type 88-8E. A MACsec secure channel is identified by a Secure Channel Identifier (SCI). A SCI is comprised of a globally unique MAC address and a Port Identifier. It is unique within the system that has been allocated that MAC address. An 8-octed SCI is appended to each 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.2 MACsec Integrity and Confidentiality	The Integrity Check Value (ICV) of the MACsec protocol data units (MPDUs) is calculated using the SAK from the destination address, source address, security tag (SecTAG), and user data (after encryption, if applicable). The ICV is encoded in the last 8 to 16 octets of the MPDU. The length of the ICV is between 8 and 16 octets, depending on the Cipher Suite. The 64 most significant bits of the 96-bit IV used in generating the ICV are the octets of the SCI, and the 32 least significant bits of the 96-bit IV are the octets of the PN. MACsec allows IPv4, IPv6, TCP, and UDP headers to be unencrypted while the rest of the frame is encrypted. The offset values and characteristics for the MACsec protected frames are: – Offset 0 is the default value. The entire MPDU payload in the frame is encrypted. – Offset 30: IPv4, TCP, and UDP headers are unencrypted. The rest of the payload is encrypted.

	– Offset 50: IPv6, TCP, and UDP headers are unencrypted. The rest of the payload is encrypted
FCS_MACSEC_EXT.3 MACsec Randomness	The TOE generates a SAK using KDF function AES-CMAC-128 or AES-CMAC-256. The KDF takes the form and inputs as follows: SAK = KDF (Key, Label, KS-nonce | MI-value list | KN, SAKlength) where 1. Key= CAK, a 128-bit value when AES-CMAC-128 is used for KDF or a 256-bit value when AES-CMAC-256 is used for KDF, 2. Label= "IEEE8021 SAK", 3. KS-nonce = a nonce of the same size as the required SAK, obtained from the TOE's approved RBG each time an SAK is generated, 4. MI-valuelist = a concatenation of Message Identified (MI) values from all live participants, 5. KN = four octets, the Key Number assigned by the Key Server as part of the KI, and 6. SAKlength = two octets representing the length of a SAK. SAKlength is an integer value (128 for a 128-bit SAK, 256 for a 256-bit SAK) with the most significant octet first.
FCS_MACSEC_EXT.4 MACsec Key Usage	Each distributed SAK is protected by AES Key Wrap (KW). The KW uses Key Encryption Key (KEK) as key input. KEK is derived from CAK. Each participant that considers itself to be the current Key Server can distribute a SAK by encoding the following information in transmitted MKPDUs: 1. The SAK protected by AES-KW, and 2. 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 the MKA Life Time has elapsed since the previous SAK was first distributed, or the Key Server's Potential Peer List is empty and the Packet Number (PN) is exhausted
FCS_MKA_EXT.1 MACsec Key Agreement	Each MACsec Key Agreement protocol data unit (MKPDU) is protected for integrity. The TOE uses a 128-bit Integrity Check value (ICV), generated by AES-CMAC using the Integrity Check value Key (ICK). The ICK is derived from the CAK using AES_CMAC. Prior to verifying the ICV, The TOE discards invalid MKPDUs in accordance with Sect. 11.11.4 of IEEE 802.1X. The MKPDUs are discarded when 1. The destination address of the MKPDU was an individual address, 2. The MKPDU is less than 32 octets in length, 3. The MKPDU is not a multiple of 4 octets in length, 4. 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 5. The CAK Name is not recognized. 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. The time outs are the following:

	1. MKA Hello Time can be configured to a value between 2.0 and 6.0 seconds. MKA Bounded Hello Timeout has a value of 0.5 seconds. The two are used per participant for periodic transmission. The time is initialized on each transmission and transmission on expiry. 2. MKA Life Time has a value of 6.0 seconds. It is used for the following: a. Per peer lifetime: initialized when adding to or refreshing the Potential Peers List or Live Peers List. Expiry causes removal from the list. b. Participant lifetime: initialized when participant created or following receipt of an MKPDU. Expiry causes participant to be deleted. c. 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.
FCS_RBG_EXT.1 Random Bit Generation	All random numbers used by the TOE are generated in accordance with the NIST Special Publication 800-90. The TOE uses the HMAC_DRBG implemented in the OpenSSL and kernel libraries. The HMAC_DRBG is used with HMAC-SHA-256. The selected HMAC_DRBG algorithm is seeded from a software-based entropy source which contains in the minimum 256 bits of entropy. An Entropy Assessment Report for the RBG is produced in a separate document.
FCS_SSHS_EXT.1 SSH Server Protocol	The TOE implements a SSH server for Trusted Channels between the TOE and a remote audit server, and for Trusted Paths between itself and remote administrators. SSH ensures that the communication over trusted channels and trusted paths is protected against unauthorized disclosure or modification. Secure connection 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. SSHv2 ensures that the transmitted data cannot be disclosed or altered. The SSH Server also supports trusted paths using SSHv2 protocol to ensures the confidentiality and integrity of remote user sessions. Remote administrators may initiate secure communication to the TOE from the SSH client of the remote management station by initiating a SSH session with the TOE. Assured identification of the parties is assured by password-based and public key-based authentication. SSHv2 protocol ensures that the data transmitted over the session cannot be disclosed or altered by unauthorized parties. The SSH server is implemented in accordance with RFCs 4251, 4252, 4253, 4254, 4344, 5656 and 6668. Conformance of the TOE with the RFCs is detailed in Sect. 6.3.1. The cryptographic algorithms used in the TOE implementation of SSH are detailed in Sect. 6.3.3.
FIA_AFL.1 Authentication Failure Management	The TOE implements a password based authentication for local users and for remote users. The authentication is implemented using the hardened Linux which is part of the TOE software. The TOE software implements the `login()` using the Pluggable Authentication Modules (PAM) Library calls. The password entered by the user is hashed, and the digest value is compared to the stored

FIA_UAU_EXT.2 Password-based Authentication Mechanism	reference value. The success or failure of the comparison is returned to `login()`. PAM is used for authentication management, account management, session management, and password management. The `login()` primarily uses the session management and password management functions of PAM. The administrator may configure the retry-options to specify the action to be taken when a remote authentication fails. The retry-options are applied to each user of the TOE. Users are identified by a username. The retry-options configurable to the administrator include the following: 1. The back-off factor: The length of delay (configurable to a value between 5 and 10 seconds) after each failed attempt before a new authentication attempt may occur. 2. The back-off threshold: The increase of the delay for each subsequent failed authentication attempt. 3. The tries-before-disconnect: The maximum number of times (configurable to a value between 1 and 10) the administrator is allowed to attempt password-based authentication through SSH before the connection is disconnected. 4. The lockout-period: The time in minutes (configurable between 1 and 43,200 minutes) before the administrator may attempt to log in to the TOE after being locked out due to the number of failed login attempts. The above concern with remote access to the TOE. Even if an account is locked to disallow remote access, the administrator may attempt a local login from the console. This ensures that the TOE is always accessible. An Administrator accessing the TOE locally may also unlock a remote Administrator account.
FIA_PMG_EXT.1 Password Management	Authentication data for the human users accessing the TOE is a password. Passwords are case-sensitive, alphanumeric strings. A password must of the minimum length. The minimum length may be configured by the administrator to be between 10 characters and 20 characters. Passwords must be composed of any combination of upper- and lower-case letters, numbers, special characters and any other standard ASCII, extended ASCII and Unicode characters. The allowed special characters are “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”.
FIA_PSK_EXT.1 Pre-Shared Key Composition	The TOE accepts pre-shared CAKs for MACsec key agreement protocols as defined by IEEE 802.1X. The TOE accepts bit-based pre-shared CAKs entered as a string of up to 64 hexadecimal characters.
FIA_UAU.7 Protected Authentication Feedback	For password authentication, the TOE software calls function `login()` which interacts with a user to request a username and password. The username and the password read from the user are used to identify and authenticate the user. The username entered by the administrator at the username prompt is echoed to the screen. There is no visual or other information presented to the used when the password is entered. This ensures that any potential eavesdropped with a visual access to the terminal the administrator uses for authenticating the TOE gains no information about the length or the content of the password.
FIA_UIA_EXT.1 User Identification and Authentication	The TOE requires users to be successfully identified and authenticated prior to granting them access to the controlled functions. The only functions the TOE allows on behalf of users prior to successful identification and authentication are the negotiation of the cryptographic protocols required for a trusted path for

<table>
<tr><td></td><td>user authentication, displaying of the access banner in the authentication window, and responding to ICMP Echo.</td></tr>
<tr><td>FMT_MOF.1/Functions Management of security functions behaviour</td><td>The administrator may configure the TOE to transmit audit records to an external syslog server. The transmission is over a secure SSHv2 connection which is initiated by the syslog server. If configured, the audit records are sent to the syslog server in real time.

The administrator may also configure the handling of audit data as detailed in FAU_GEN.1 and FAU_STG_EXT.1.

The audit function is stopped, and a log entry indicating exhaustion of the file system generated when the file capacity is exhausted (see FAU_STG_EXT.1). The behavior is not configurable by the administrator.</td></tr>
<tr><td>FMT_MOF.1/Services Management of security functions behaviour</td><td>The TOE implements a SSH Server to which the administrators may connect from a remote syslog server of from a remote management station. The administrator may also terminate the SSH session at any time. This allows the administrator to start and stop the trusted channels and trusted paths of the TOE.</td></tr>
<tr><td>FMT_MTD.1/CoreData Management of TSF Data</td><td>The TOE implements human user authentication using passwords. Each user is identified with a username and authenticated with a password. Access to the management functions of the TOE is only granted to successfully identified and authenticated users assigned to the role Security Administrator. The authentication function of the TOE ensures that inactive sessions are terminated, authentication failures are handled in a manner that prevents password guessing, passwords may not be accessed in the file system, and the passwords are not echoed on the terminal when a user is being authenticated. Any remote management session is protected with SSHv2. These measures jointly ensure that the access to the management functions is only granted to legitimate administrators, and that the non-administrative users are effectively prevented from accessing the management functions.</td></tr>
<tr><td>FMT_MTD.1/CryptoKeys Management of TSF data</td><td>The TOE allows successfully authenticated administrators to perform the following management functions on the cryptographic keys:
– Manage the threshold for SSH rekeying,
– Generation of SSH keys,
– Configuration of pre-shared MACsec CAKs,
– Generate a MACsec PSK and install it in the device.
– Manage the Key Server to create, delete, and activate MKA participants, and
– Enable, disable, and delete a MACsec PSK-based CAK

The cryptographic keys used by the TOE and the mechanisms available for the administrator to erase them is detailed in Sect. 6.3.2.</td></tr>
<tr><td>FMT_SMF.1 Specification of Management Functions

FMT_SMF.1/MACSEC Specification of Management Functions (MACsec)</td><td>The TOE implements a Command Line Interface (CLI) which allows the administrators to manage the TOE. The CLI may be accessed locally from console or from a remote management station over a SSHv2 connection. The entire CLI is accessible to all administrator, whether accessing the TOE locally or remotely. The TOE prevents access to the CLI by unauthenticated and unauthorized users.</td></tr>
</table>

	The TOE implements the following management functions fully detailed in the security guidance: – Configuring the access banner, – Configuring the session inactivity time before session termination, – Updating the TOE software, and verifying the update using digital signature capability prior to installation, – Starting and stopping services – Configuring the local audit behaviour, – Managing the cryptographic keys, – Configuring the thresholds for SSH rekeying, – Re-enabling a locked Administrator account, – Setting the time used for time-stamps, – Configure the reference identifier for the peer, – Configuring the authentication failure parameters, and – Managing the SSH key databases. MACsec may also be managed through the CLI. The following additional management functions allow the administrator to specifically manage the MACsec: – Manage a PSK-based CAK and install it in the device, – Manage the key server to create, delete, and activate MKA participants, – Specify the lifetime of a CAK, and – Enable, disable, or delete a PSK-based CAK.
FMT_SMR.2 Restrictions on Security Roles	The only role maintained by the TOE is a Security Administrator. Only users accessing the TOE from user accounts assigned to the Security Administrator role are granted the right to administer the TOE. Each user account has attributes user identity (username), authentication data (password) and role (privilege) assigned to it. The role Security Administrator is associated with a login class "security-admin". The security-admin logic class is assigned the necessary privileges which permit the users to perform all management functions of the TOE. Security Administrators may administer the TOE locally from system console or remotely over a SSHv2 trusted path using the SSHv2 protocol.
FPT_APW_EXT.1 Protection of Administrator Passwords	The passwords of the users are hashed when stored in the local password file. Hashing may be configured to be with SHA-256 or SHA-512. The CLI implements no functions for accessing the passwords directly. SHA-256 and SHA-512 are cryptographically secure. Even if gaining access to the hashed passwords, the has no practical means of recovering the password from the hash value. Authentication data for public key-based authentication is stored in a directory owned by the user. The directory typically has the same name as the user. The directory contains the files `.ssh/authorized_keys` and`.ssh/authorized_keys2` which are used for SSH public key authentication. The CLI allows no direct access to the files and the authentication data may only be accessed through the CLI commands for managing the keys, or by the privileged processes implementing the SSH Server. They are not directly accessible to the administrators.

FPT_CAK_EXT.1 Protection of CAK Data	The CAK Certificate Chain is stored AES-encrypted in a configuration file using the System Master Password. The System Master Password is stored in the non-volatile memory of the TOE. The CLI does not implement any commands which would grant the administrator access to the CAP Certificate Chain or the System Master Password. The administrators also do not have root access to the file systems of the TOE. This protects the CAK from any unauthorized access.
FPT_RPL.1 Replay Detection	The TOE implements measures to protect from MACsec replay attacks at the control plane and at the data plane. To protect against replay attacks in the control plane, each participant in the protocol chooses a random 96-bit member identifier (MI). When a MKA exchange commences, the MI is used together with a 32-bit message number (MN). MN is initialized to 1 and incremented with each MKPDU transmitted. The combination of MI and MN is used for detecting any attempt to replay a MKPDU message. The TOE implements data plane replay functionality to ensure that a man-in-the middle attacker cannot replay a snooped packet or reuse a packet number. As the TOE does not support bounded receive delay functionality, 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 the replay-window-size is set to zero, replays are not permitted and shall not be used when out of ordering is detected or expected.
FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)	The CLI implemented by the TOE does not include commands for viewing the cryptographic keys. The TOE enforces kernel-level file access rights to the key containers. The access rights granted by the TOE limit access to the contents of cryptographic key containers only to the processes with cryptographic rights and to the shell users with root permission. As security administrators do not have root permission to the Junos OS, the measures restrict access to the contents of the key containers to authorized processes only.
FPT_STM_EXT.1 Reliable Time Stamps	The TOE implements a clock based on a hardware time stamp counter. The time may be set by the administrator. Synchronizing the time with a NTP Server is not supported. The clock may be used for generating real-time time stamps and counters indicating the time from a specific event. The clock is used by the TOE to produce a time stamp for each audit record generated by the TOE, to implement inactivity timers for the administrative sessions, to implement the periods on which a user may not attempt re-authentication after a failed authentication attempt, and to implement protocol timers required for triggering re-keying or termination of a protocol session.
FPT_FLS.1 Failure with Preservation of Secure State FPT_TST_EXT.1 TSF Testing	The TOE runs the following set of self-tests when powered on to verify the correct operation of the TOE software: 1. Power on test to determine that the boot-device responds and performs a memory size check to confirm the amount of available memory. 2. File integrity test to verify each mounted signed package and to assert that system files have not been tampered with. To test the integrity of the firmware, the SHA-1 fingerprints of the executables and other immutable files are regenerated and validated against the reference fingerprints contains in the manifest file.

	3. Crypto integrity test to check the integrity of cryptographic keys and major CSPs. 4. Authentication error to verify that veriexec is enabled and operates as expected using `/opt/sbin/kats/cannot-exec.real`. 5. Kernel, libmd, OpenSSL, QuickSec, SSH tests to verify the output from known answer tests for the cryptographic algorithms. The TOE only executes binaries supplied by Juniper Networks. Within the package containing the TOE software, each Junos OS firmware image includes fingerprints of the executables and other immutable files. The TOE will not execute any binary without validating the registered fingerprint. This protects the TOE from unauthorized firmware which might compromise the integrity of the TOE. The self-tests ensure that only authorized executables are allowed to run and ensure the correct operation of the TOE. In case of a corrupt state or a failure in a self-test, the TOE will panic. The event will be logged, the TOE will cease processing network traffic and CLI commands, and restart. When the TOE restarts, the boot process shall not succeed without passing each self-test. This constitutes the automatic recovery and self-test behavior of the TOE
FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour FPT_TUD_EXT.1 Trusted Update	Administrators of the TOE may query the current version of the TOE firmware using the CLI command `show version`. if a new version of the TOE firmware is available, the administrators may initiate an update of the TOE firmware. The TOE does not allow partial updates. The administrator must upgrade to the entire new release. Updates are downloaded and applied manually. The TOE does not implement automatic updates. The installable firmware package containing an update to the TOE software has a digital signature attached. The digital signature is computed using ECDSA (P-256) with SHA-256 in the development environment of the TOE. The TOE checks the digital signature and only proceeds with the installation of the verification succeeds. The TOE maintains a set of fingerprints (i.e. SHA-1 digests) for executable files and other files which should be immutable. The manifest file is digitally signed using the Juniper package signing key in the development environment The signature is verified by the TOE. The fingerprint loader will only process a manifest for which it can verify the signature. Without a valid digital signature an executable will not be executed. When the command is issued to install an update, the manifest file for the update is verified and stored, and each executable/immutable file is verified before being executed. If any of the fingerprints in an update fails the verification, the upgrade will fail, and the TOE will use the last known verified image instead.
FTA_SSL.3 TSF-initiated Termination FTA_SSL_EXT.1 TSF-initiated Session Locking	The administrator may configure the TOE to terminate each local and remote user session after a period of inactivity. The TOE implements a clock and generates an instance of a counter for each user to track the clock cycles since last activity. The count is reset each time the TOE detects activity on the user session. When the instance of a counter reaches the number of clock cycles equating to the configured period of inactivity, the user session is locked out.

	the Administrator may configure the inactivity period. The default value is 30 seconds. When a user is locked out, the TOE overwrites the display device and makes the current contents unreadable. The session is also terminated to prevent any further interaction with the TOE until a successful re-authentication.
FTA_SSL.4 User-initiated Termination	Each user sessions, whether local or remote, can be terminated by the user. The user may log out of an existing local or remote session by issuing a `logout` command. When the command is issued, the user exits the session, and the TOE makes the current contents unreadable. No user activity can take place until a successful re-authentication.
FTA_TAB.1 Default TOE Access Banners	The administrator may configure an access banner to be displayed at each local and remote authentication exchange. The banner may provide warnings against unauthorized access to the TOE and any other information that the administrator wishes to communicate.
FTP_ITC.1 Inter-TSF Trusted Channel FTP_ITC.1/MACSEC Inter-TSF Trusted Channel (MACsec Communications) FTP_TRP.1/Admin Trusted Path	The TOE implements trusted channels with SSHv2 and MACsec and trusted paths with SSHv2. SSHv2 may be used by a remote syslog server to establish a secure connection with the TOE at the network layer. The secure connection may be used by the TOE to forward audit records to the syslog server for storage and further processing. MACsec may be used by the TOE to establish a secure link layer connection to an authenticated MACsec peer for secure communication. MACsec connection may be established by the TOE or by the MACsec peer. The TOE allows for remote administration of the TOE. The administrator may use a SSHv2 client of a remote management station to connect to the TOE. Upon successful authentication, the TOE establishes a SSHv2 session with the SSH client of the remote management station and uses that for securing all administrative commands and responses thereof.

6.2 Fulfillment of the Security Assurance Requirements

To fulfill the Security Assurance Requirements, the developer implements a set of security assurance measures. Some assurance classes are fulfilled by the evaluator of the TOE. The security assurance measures implemented by the developer and evaluator of the TOE are described in Table 18.

Table 18 Fulfillment of the Security Assurance Requirements

Security Assurance Requirement	Fulfilment
Security Target	The developer authors a Common Criteria Security target for the Target of Evaluation. The Security Target implements all assurance components required by the Base-PP. The Security Target includes – A ST Introduction which provides a ST Reference, a TOE Reference, a TOE Overview, and a TOE Description.

	– Conformance Claims stating exactly the conformance to the Common Criteria and the Protection Profiles, Protection Profile Modules and Protection Profile Configurations the Security Target and the Target of Evaluation claim conformance to. – A Security Problem Definition which is a statement of Threats, Assumptions and Organizational Security Policies applicable to the TOE. – A statement of the security objectives for the TOE. The Base-PP only defines security requirements for the operational environment of the TOE, but the Security Target also states the security requirements for the TOE drawn from the PP-Module. – Extended Components Definition and the statement of the security requirements state exactly the Security Functional Requirements and the Security Assurance Requirements the TOE fulfills. – TOE Summary Specification which describes for each Security Functional Requirement how the TOE fulfills that Security Functional Requirement.
Functional Specification	Included in the TOE Summary Specification, the developer provides all information required for a basic functional specification of the TOE.
Security Guidance	Attached to the TOE and included in the physical scope of the TOE is a Common Criteria Guidance Supplement for the TOE. The Guidance Supplement gives guidance to the user of the TOE in the secure installation and preparation of the TOE so that the TOE is in an initial secure state. The Guidance Supplement also provides guidance to the user of the TOE so that the TOE always remains in a secure state when the guidance is followed.
Life Cycle Support	The developer labels the TOE with the unique identifier. The label may be examined by the user of the TOE to ensure that the correct version of the TOE is used. When the TOE software is updated, the label of the TOE is updated accordingly. The TOE label is included in the configuration list of the TOE to ensure that the evaluator can be assured of evaluating the intended version of the TOE.
Independent Testing	The evaluator carries out a set of independent tests on the TOE. The independent tests complement the functional testing carried out by the developer and ensure that the TOE passes each applicable test required for conformance with the Base-PP, PP-Module and Functional Package. The evaluator documents the testing in accordance with the requirements stated in the Base-PP, the PP-Module, the Functional Package and the Common Criteria evaluation and certification scheme followed.
Vulnerability Assessment	The evaluator carries out a vulnerability survey to determine that there are no obvious vulnerabilities in the TOE which could be practically exploited by the threat agents. The evaluator documents the vulnerability survey in accordance with the requirements stated in the Base-PP, the PP-Module, the Functional Package and the Common Criteria evaluation and certification scheme followed.

6.3 Cryptographic Details and CAVP References

This section provides additional details on the cryptographic algorithms and protocols implemented by the TOE.

6.3.1 SSH RFC Conformance

The conformance of the TOE implementation of SSH to the applicable RFCs is given in Table 19.

Table 19 RFCs Applicable to SSH

RFC	RFC Summary	Implementation
RFC 4251	The Secure Shell (SSH) Protocol Architecture	Host Keys: The TOE uses a 256-bit ECDSA Host Key for SSHv2. The host key is generated on the initial setup of the TOE. It can be de-configured via the CLI. De-configuration deletes the key and makes it unavailable for a connection establishment. The key is generated randomly to be unique to each TOE instance. The TOE presents the SSH client with its public key and the client matches the presented 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. The TOE 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 but has no X11 libraries or applications and prohibits X11 forwarding. Confidentiality: The TOE does not accept the "none" cipher but supports AES-CBC-128, AES-CBC-256, AES-CTR-128, AES-CTR-256 encryption algorithms for protection of data over SSH. The keys generated in accordance with "ssh-rsa", "rsa-sha2-256", "rsa-sha2-512", "ecdsa-sha2-nistp256", "ecdsa-sha2-nistp384" or "ecdsa-sha2-nistp521" are used for public-key based device authentication. For ciphers whose blocksize is greater or equivalent to 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 can be configured between 51,200 and 4,294,967,295 (2^32-1) bytes and the time-limit must be between 1 and 1440 minutes. In the evaluated configuration the time-limit must be set within 1 and 60 minutes. Denial of Service: When a SSH connection is brought down, the TOE does not attempt to re-establish it. Ordering of Key Exchange Methods: The key exchange algorithms supported by the TOE include and are ordered as follows: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group14-sha1.

		Debug Messages: The TOE does not allow debugging messages via the CLI. End Point Security: The TOE permits port forwarding. Proxy Forwarding: The TOE permits proxy forwarding. X11 Forwarding: The TOE does not support X11 forwarding.
RFC 4252	The Secure Shell (SSH) Authentication Protocol	Authentication Protocol: The TOE does not accept the "none" authentication method and replies with a list of permitted authentication methods. The TOE implements a timeout period of 30 seconds for authentication of the SSHv2 protocol and allows for 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 also does not allow authentication requests for a non-existent username to succeed. It sends back a disconnect message as it would for failed authentications. This prevents enumeration of valid usernames. 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 authentication methods (public key and password) for users. Password Authentication Method: The TOE supports password authentication. Expired passwords cannot be used for authentication. Host-Based Authentication: The TOE does not support host-based authentication methods.
RFC 4253	The Secure Shell (SSH) Transport Layer Protocol	Encryption: The TOE allows the following methods for the encryption of SSH sessions: aes128-cbc and aes256-cbc, aes128-ctr, and 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: The TOE drops packets greater than 263,000 bytes in SSH transports and terminates the connection. Data Integrity: The TOE does permit negotiation of HMAC-SHA1 in each direction for SSH transport. Key Exchange: The TOE does support diffie-hellman-group14-sha1. Key Re-Exchange: The TOE performs a re-exchange when SSH_MSG_KEXINIT is received.
RFC 4254	Secure Shell (SSH) Connection Protocol	Multiple channels: The TOE assigns each channel a number as detailed in RFC 4251. Data transfers: The TOE supports a maximum window size of 256,000 bytes for data transfer. Interactive sessions: The TOE only supports interactive sessions that do not involve X11 forwarding.

		Forwarded X11 connections: Forwarded X11 connections are not supported by the TOE. Environment variable passing: The TOE only sets variables once the server process has dropped privileges. Starting shells/commands: The TOE allows only one request for a shell, application program, or command 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: Port forwarding is fully supported by the TOE.
RFC 4344	Secure Shell (SSH) Transport Layer Encryption Modes	The TOE implements the recommended modes of operation aes128-ctr and aes256-ctr. It does not implement the recommended modes of operation aes192-ctr and 3des-ctr. The TOE does also not implement any of the optional modes.
RFC5656	Secure Shell (SSH) Transport Layer Encryption Modes	ECDH Key Exchange: The TOE implements the key exchange methods ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521. The SSH client matches the key returned by the TOE against its `known_hosts` list of keys. Hashing: The TOE implements SHA-256 and SHA-512 algorithms. The message digest size is either 256 or 512 bits. Required Curves: Each required curve is implemented: ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521. None of the Recommended Curves are supported.
RFC 6668	sha2-Transport Layer Protocol	Both the recommended algorithm hmac-sha2-256 and the optional algorithm hmac-sha2-512 are implemented for SSH transport.
RFC 8308 Section 3.1	Extension Negotiation in the Secure Shell (SSH) Protocol	The extension negotiation is implemented for RSA with SHA-256 and for RSA with SHA-512.
RFC 8332	Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol	The TOE implements both rsa-sha2-256 and rsa-sha2-512.

6.3.2 Zeroization of Cryptographic Keys and Critical Security Parameters

The timing and method of the zeroization of the cryptographic keys and critical security parameters (CSP) used by the TOE is given in Table 20.

Table 20 Timing and Method of the Zeroization of Cryptographic Keys and Critical Security Parameters

Key/CSP	Storage Format	Storage Location	Zeroization Method

SSH Private Host Key	Plaintext	Non-volatile memory	When the TOE is recommissioned, the config files (including SSH keys) are erased by the administrator using the `request vmhost zeroize no-forwarding` CLI command
	Plaintext	Volatile memory	`free()` performed by the TOE software at session termination
SSH Session Key	Plaintext	Volatile memory	`free()` performed by the TOE software at session termination
User Password	Plaintext when entered	Volatile memory	`free()` performed by the TOE software at the completion of the user authentication
	Hashed when stored	Non-volatile memory	When the TOE is recommissioned, the config files (including user passwords in the password file) are erased by the administrator using the `request vmhost zeroize no-forwarding` CLI command
RNG State	Plaintext	Volatile memory	Overwritten by the kernel of the TOE with zeros at reboot
MACsec CAK	AES encrypted with System Master Password	Config file	Zeroized by the administrator by issuing the `request vmhost zeroize no-forwarding` CLI command
MACsec SAK	Plaintext	Volatile memory	`free()` performed by the TOE software at session termination
MACsec KEK	Plaintext	Volatile memory	`free()` performed by the TOE software at session termination
MACsec ICK	Plaintext	Volatile memory	`free()` performed by the TOE software at session termination
System Master Password	Plaintext	Non-volatile memory	Zeroized by the administrator by issuing the `request vmhost zeroize no-forwarding` CLI command

6.3.3 Cryptographic Algorithms Used for SSH and MACsec

The cryptographic algorithms and methods used by the TOE are summarized in Table 21.

Table 21 Cryptographic Algorithms and Methods Used by the TOE

Protocol	Key Establishment	Authentication	Encryption	Data Integrity
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	AES CTR 128 AES CTR 256 AES CBC 128 AES CBC 256	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-512

		ecdsa-sha2-nistp521		
MACsec	N/A	GMAC	AES GCM 128 AES GCM 256	AES GCM 128 AES GCM 256
MKA	AES Key Wrap (CMAC)	Static-CAK (preshared)	AES CBC 128 AES CBC 256	AES CMAC

6.3.4 CAVP Certificate References

The TOE implements a rich set of cryptographic functions used to protect communications and the integrity of the security functions. Each cryptographic function of the TOE is CAVP validated. The CAVP certificate references, organized by the applicable Security Functional Component, are given in Table 22.

The cryptographic algorithms are implemented in the following parts of the TOE:

- QSMX: Quicksec (Inside Secure) for Junos OS 23.4R1
- OSMX: OpenSSL for Junos OS 23.4R1 (based on OpenSSL 1.1.1n)
- LIMX: LibMD for Junos OS 23.4R1 (created from same sources as OpenSSL, namely 1.1.1n)
- KEMX: Kernel for Junos OS 23.4R1 (based on FreeBSD-12 Stable release)
- MSMX: MACsec firmware for Junos OS 23.4R1
- LC480: Linecard LC480 with Marvell Alaska C PHYs chipset
- LC9600: Linecard LC9600 with Juniper Trip 6 chipset

Each CAVP Certificate is applicable to each variant of the TOE.

Table 22 CAVP Certificate References

Security Functional Component	Function/Algorithm	Capabilities	Part	CAVP Reference
FCS_CKM.1	Asymmetric Key Generation for SSH	RSA-2048 RSA-4096 ECC P-256 ECC P-384 ECC P-521	OSMX	A5633
FCS_CKM.2	ECC Key Establishment for SSH	ECC P-256, SHA-256 ECC P-384, SHA-384 ECC P-521, SHA-512	OSMX	A5633
	FFC Key Establishment for SSH	ECC P-256, SHA-256 ECC P-384, SHA-384 ECC P-521, SHA-512	OSMX	N/A - Tested against known good implementation

FCS_COP.1/DataEncryption	SSH Encryption and Decryption	AES-CBC-128 AES-CBC-256 AES-CTR-128 AES-CTR-256	OSMX	A5633
FCS_COP.1/SigGen	Digital Signature Computation and Verification	ECC P-256, SHA-256 ECC P-384, SHA-384 ECC P-521, SHA-512 RSA-2048, SHA-256 RSA-4096, SHA-256	OSMX	A5633
FCS_COP.1/Hash	Message Digest Computation	SHA-1 SHA-256 SHA-384 SHA-512	OSMX	A5633
		SHA-1 SHA-256 SHA-384 SHA-512	KEMX	A5058
		SHA-1 SHA-256	LIMX	A5059
FCS_COP.1/KeyedHash	Keyed Hash Message Digest Computation	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-512	OSMX	A5633
		HMAC-SHA-1 HMAC-SHA-256	KEMX	A5058
		HMAC-SHA-1 HMAC-SHA-256	LIMX	A5059
FCS_COP.1/CMAC	MACsec Keyed Hashing	AES-CMAC-128 AES-CMAC-256	MSMX	A5060
FCS_COP.1/MACSEC	MACsec Data Encryption and Decryption	AES-GCM-128 AES-GCM-256	LC480	A5651
		AES-GCM-128 AES-GCM-256	LC9600	A4664
	MACsec Key Wrap	AES-KW-128 AES-KW-256	MSMX	A5060
FCS_MACSEC_EXT.2		AES-GCM-128	LC480	A5651

	MACsec Data Encryption and Decryption	AES-GCM-256		
		AES-GCM-128 AES-GCM-256	LC9600	A4664
	Random Nonce Generation	HMAC-SHA-256	KEMX	A5058
FCS_MACSEC_EXT.3	Random Nonce Generation	HMAC-SHA-256	KEMX	A5058
	MACsec Key Derivation	ASE_CMAC-128 AES-CMAC-256	MSMX	A5058
FCS_MACSEC_EXT.4	MACsec Key Wrap	AES-KW-128 AES-KW-256	MSMX	A5060
FCS_RBG_EXT.1	HMAC-DRBG	HMAC-SHA-256	KEMX	A5058
FCS_SSHS_EXT.1	SSH Data Integrity	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-512	OSMX	A5633
		HMAC-SHA-1 HMAC-SHA-256	KEMX	A5058
		HMAC-SHA-1 HMAC-SHA-256	LIMX	A5059
	SSH Key Establishment	ECDH-SHA2-NISTP256 ECDH-SHA2-NISTP384 ECDH-SHA2-NISTP521	OSMX	A5633
		DH-Group-14-SHA256	OSMX	N/A - Tested against known good implementation
	SSH Peer Entity Authentication	SSH-RSA RSA2048-SHA2-256 RSA4096-SHA2-256 ECDSA-SHA2-NISTP256 ECDSA-SHA2-NISTP384 ECDSA-SHA2-NISTP512	OSMX	A5633
	SSH Data Encryption	AES-CTR-128 AES-CTR-256	OSMX	A5633

		AES-CBC-128 AES-CBC-256		
FPT_TUD_EXT.1	Digital Signature Verification	ECDSA P-256, SHA-256 ECDSA P-384, SHA-384 ECDSA P-521, SHA-512	OSMX	A5633

7 Acronyms

AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
C	Celsius
CA	Connectivity Association
CAK	Connectivity Association Key
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Scheme
CEM	Common Evaluation Methodology
CKN	Connectivity Association Key Name
CLI	Command Line Interface
cm	Centimeter
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
DHCP	Dynamic Host Configuration Protocol
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
FFC	Finite Field Cryptography
EAP	Extensible Authentication Protocol
EAPOL	EAP Over LAN
ECDSA	Elliptic Curve Digital Signature Algorithm
EMI	Electromagnetic Interference
F	Fahrenheit
FIPS	Federal Information Processing Standard
FIPS PUB	FIPS Publication
FTP	File Transfer Protocol

GB	Giga-Bit
GbE	Giga-bit Ethernet
GCM	Galois Counter Mode
GHz	Giga-Hertz
HMAC	Hash-Based Message Authentication Code
ICK	Integrity Check Value Key
ICMP	Internet Control Message Protocol
ICV	Integrity Check Value
IEEE	Institute of Electrical and Electronics Engineers
IFD	Interface Device
in	Inch
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISO	International Organization for Standardization
IT	Information Technology
IV	Initialization Vector
KaY	Key Agreement Entity
KDF	Key Derivation Function
KEK	Key Encryption Key
kg	kilogram
KN	Key Number
KVN	Kernel-based Virtual Machine
KW	Key Wrap
LAN	Local Area Network
lb	pound [From libra pondo -> libra -> lb]
LED	Light Emitting Diode
LLDP	Link Layer Discovery Protocol
LSR	Label-Switching Router
MKA	MACsec Key Agreement
MAC	Media Access Control

	or: Message Authentication Code
MACsec	Media Access Control Security
MI	Message Identifier
MKA	MACsec Key Agreement
MKPDU	MACsec Key Agreement Protocol Data Unit
MPDU	MACsec Protocol Data Unit
MN	Message Number
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
OS	Operating System
OSP	Organizational Security Policy
PAE	Port Access Entity
PAM	Pluggable Authentication Modules
PFE	Packet Forwarding Engine
PKCS	Public Key Cryptography Standard
PSK	Pre-Shared Key
PSS	Improved Probabilistic Signature Scheme
RCB	Routing and Control Board
RE	Routing Engine
RSA	Rivest-Shamir-Adleman
RSASSA	RSA Signature Scheme with Appendix
RFC	Request For Comments
RBG	Random Bit Generator
SA	Secure Association
SAK	Secure Association Key
SCI	Secure Channel Identifier
SD	Supporting Document
SDN	Software-Defined Networking
secTAG	[MACsec] Security Tag
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SSD	Solid State Drive

SSH Secure Shell

SSL Secure Socket Layer

STP Spanning Tree Protocol

Tbps Tera-bits Per Second

TLS Transport Layer Security

TSF TOE Security Function

TCP Transmission Control Protocol

UDP User Datagram Protocol

XPN Extended Packet Numbering

8 References

CC *Common Criteria for Information Technology Security Evaluation*

- Part 1: Introduction and general model, April 2017, Version 3.1 Revision 5, CCMB-2017-04-001
- Part 2: Security functional components, April 2017, Version 3.1 Revision 5, CCMB-2017-04-002
- Part 3: Security assurance components, April 2017, Version 3.1 Revision 5, CCMB-2017-04-003

CEM *Common Methodology for Information Technology Security Evaluation, Evaluation methodology, April 2017, Version 3.1 Revision 5, CCMB-2017-04-004*

NIAP Policy Letter #5 Applicability and Relationship of NIST Cryptographic Algorithm Validation Program (CAVP) and Cryptographic Module Validation Program (CMVP) to NIAP's Common Criteria Evaluation and Validation Scheme (CCEVS)

- Addendum #1 – Frequently Asked Questions for NIAP Policy #5
- Addendum #2 – CAVP Mapping, Version 2.0

Base-PP collaborative Protection Profile for Network Devices Version: 2.2e, Date: 23-March-2020 (cpp_nd_v2.2e)

PP Module PP-Module for MACsec Ethernet Encryption Version: 1.0, 2023-03-02 (mod_macsec_v1.0)

PP-Configuration PP-Configuration for Network Devices and MACsec Ethernet Encryption Version: 1.0, 2023-03-29 (CFG_NDcPP-MACsec_V1.0)