Juniper Networks

Junos OS 24.2R2 for EX4400

Assurance Activity Report

Version 1.1

May 2026

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Author	Reviewer	Description
0.1	03/25/2025	S. Bennett		Initial Draft.
0.2	05/30/2025	S. Bennett		Updated per OR responses
1.0	03/30/2026	S. Bennett	C. Cantlon	Finalization
1.1	05/27/2026	S. Bennett		Address ECRs

Table of Contents

1 INTRODUCTION 5
1.1 Evaluation Identifiers 5
1.2 Evaluation Methods 5
1.3 Reference Documents 7

2 TESTING OVERVIEW 8
2.1 TOE Overview 8
2.2 TOE Models 8

3 EVALUATION ACTIVITIES FOR MANDATORY SFRS 11
3.1 Security Audit (FAU) 11
3.2 Cryptographic Support (FCS) 18
3.3 Identification and Authentication (FIA) 39
3.4 Security management (FMT) 41
3.5 Protection of the TSF (FPT) 47
3.6 TOE Access (FTA) 54
3.7 Trusted path/channels (FTP) 57

4 EVALUATION ACTIVITIES FOR OPTIONAL REQUIREMENTS 61
4.1 Security Audit (FAU) 61

5 EVALUATION ACTIVITIES FOR SELECTION-BASED REQUIREMENTS 63
5.1 Cryptographic Support (FCS) 63
5.2 Identification and Authentication (FIA) 67
5.3 Protection of the TSF (FPT) 71
5.4 Security management (FMT) 71
5.5 TOE Access (FTA) 78

6 EVALUATION ACTIVITIES FOR SECURITY ASSURANCE REQUIREMENTS 79
6.1 ASE: Security Target Evaluation 79
6.2 ADV: Development 80
6.3 AGD: Guidance Documents 81
6.4 ALC: Life-cycle Support 85
6.5 ATE: Tests 86
6.6 Vulnerability Assessment 87

7 EVALUATION ACTIVITIES FOR MANDATORY SFRS (PKG_SSH) 90
7.1 Cryptographic Support (FCS) 90

8 OPTIONAL REQUIREMENTS (PKG_SSH) 100
8.1 Strictly Optional Requirements 100
8.2 Objective Requirements 100
8.3 Implementation-based Requirements 100

9 EVALUATION ACTIVITIES FOR SELECTION-BASED REQUIREMENTS (PKG_SSH) 101
9.1 Cryptographic Support (FCS) 101

10 PP-MODULE FOR MACSEC ETHERNET ENCRYPTION (MOD_MACSEC) 103
10.1 Modified SFRs 103

11 TOE SFR EVALUATION ACTIVITIES (MOD_MACSEC) 104
11.1 Security Audit (FAU) 104
11.2 Cryptographic Support (FCS) 104
11.3 Identification and Authentication (FIA) 117
11.4 Security Management (FMT) 119
11.5 Protection of the TSF (FPT) 121
11.6 Trusted Path/Channels (FTP) 124

12 EVALUATION ACTIVITIES FOR OPTIONAL SFRS (MOD_MACSEC) 125

13 EVALUATION ACTIVITIES FOR SELECTION-BASED SFRS (MOD_MACSEC) 126
14 EVALUATION ACTIVITIES FOR OBJECTIVE SFRS (MOD_MACSEC) 127
15 EVALUATION ACTIVITIES FOR IMPLEMENTATION-BASED SFRS (MOD_MACSEC) 128
16 EVALUATION ACTIVITIES FOR SARS (MOD_MACSEC) 129

1 Introduction

1 This Assurance Activity Report (AAR) documents the evaluation activities performed by Lightship Security for the evaluation identified in Table 1. The AAR is produced in accordance with National Information Assurance Partnership (NIAP) reporting guidelines.

1.1 Evaluation Identifiers

Table 1: Evaluation Identifiers

Scheme	NIAP Common Criteria Evaluation and Validation Scheme
Evaluation Facility	Lightship Security
Developer/Sponsor	Juniper Networks, Inc.
TOE	Junos OS 24.2R2 for EX4400
Security Target	Junos OS 24.2R2 for EX4400 Security Target, v1.1
Protection Profile	PP-Configuration for Network Devices and MACsec Ethernet Encryption, Version 2.0 (CFG_NDcPP-MACsec_V2.0) [CFG] Base PP: collaborative Protection Profile for Network Devices, Version 3.0e (CPP_ND_V3.0E) [NDcPP] PP-Module: PP-Module for MACsec Ethernet Encryption, Version 1.0 (MOD_MACSEC_V1.0) [MACSEC-MOD] Functional Package for Secure Shell (SSH), Version 1.0 (PKG_SSH_V1.0) [SSH-FP]

1.2 Evaluation Methods

2 The evaluation was performed using the methods and standards identified in Table 2.

Table 2: Evaluation Methods

Evaluation Criteria	CC v3.1R5
Evaluation Methodology	CEM v3.1R5 CC and CEM addenda: Exact Conformance, Selection-Based SFRs, Optional SFRs, v0.5
Supporting Documents	Evaluation Activities for Network Device cPP, Version 3.0e, 06-December-2023 [ND-SD] Supporting Document Mandatory Technical Document PP-Module for MACsec Ethernet Encryption, Version: 1.0 [MACSEC-SD]

Table 3: Technical Decisions

Technical Decisions	Applicable
TD0682 - Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	Applicable
TD0695 - Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.	Applicable
TD0728 - Corrections to MACSec PP-Module SD	Applicable
TD0732 - FCS_SSHS_EXT.1.3 Test 2 Update	Applicable
TD0746 - Correction to FPT_RPL.1 Test 25	Applicable
TD0777 - Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Applicable
TD0803 - Clarification for Configurable MACsec CKN Length	Applicable
TD0816 - Clarity for MACsec Self Test Failure Response	Applicable
TD0825 - Correction to IEEE 802.1X Reference	Applicable
TD0826 - Aligning MOD_MACSEC_V1.0 with CPP_ND_V3.0E	Applicable
TD0836 – NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1	Applicable
TD0840 - Alignment of Test 22.1 to FMT_SMF.1/MACSEC	Applicable
TD0868 -NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	Not Applicable – TOE does not claim IPSec functionality
TD0870 - Security Objectives Rationale for MOD_MACSEC_V1.0	Applicable
TD0879 - NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E	Applicable
TD0880 - NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1	Applicable
TD0881 - Correction to MN Usage for FPT_RPL.1 Test	Applicable
TD0882 - MACsec Data Delay Protection, Key Agreement, and Conditional Support for Group CAK	Applicable
TD0884 - Expansion of Permitted EtherTypes in FCS_MACSEC_EXT.1.4	Applicable
TD0886 - Clarification to FAU_STG_EXT.1 Test 6	Applicable
TD0889 - Correction For Tests Incorrectly Requiring Group MACsec	Applicable

Technical Decisions	Applicable
TD0891 - Correlation of Implicitly Satisfied Requirements when CPP_ND_V3.0E is the Base-PP	Applicable
TD0899 - NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2	Not Applicable – TOE does not claim TLS functionality
TD0900 - NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	Applicable
TD0909 - Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	Applicable
TD0921 - NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment	Applicable
TD0923 - NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2	Applicable
TD0939 - Updated Conformance Claims for MOD_MACSEC	Applicable
TD0967 - Allowance of Kex-strict in PKG_SSH_V1.0	Applicable
TD0973 - NIT Technical Decision: FCS_(D)TLSS_EXT.1.3 Test 2 DHE Ciphersuite Conditionality	Not Applicable – TOE does not claim TLS functionality
TD0990 - NIT Technical Decision: CTR_DRBG in FCS_RBG_EXT.1.2	Applicable

1.3 Reference Documents

Table 4: List of Reference Documents

Ref	Document
[ST]	Junos OS 24.2R2 for EX4400 Security Target, v1.1
[AGD]	Juniper Networks Junos OS Common Criteria Evaluated Configuration Guide for EX4400-24MP, EX4400-24P, EX4400-24T, EX4400-24X, EX4400-48F, EX4400-48MP, EX4400-48P, and EX4400-48T Devices, Release 24.2R2, 2026-03-27

2 Testing Overview

2.1 TOE Overview

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

4 The TOE provides the following security functions:

- Security Audit. The TOE generates audit records of user and administrator actions. The TOE includes the user identity in audit events resulting from actions of identified users. The TOE is capable of sending logs in real-time to an external syslog server via SSHv2.
- Cryptographic Support The TOE implements a cryptographic module. The cryptographic module has the ability to generate and destroy cryptographic keys. Relevant Cryptographic Algorithm Validation Program (CAVP) certificates are shown in section NIAP Policy 5.
- Identification and Authentication. The TOE ensures that all users must be identified and authenticated before accessing its functions and data.
- Security Management. The TOE provides a suite of security management functions that is available to authorized administrators in support of configuring and maintaining the TOE.
- Protection of the TSF. The TOE performs a suite of self-tests to ensure the correct operation and enforcement of its security functions including software integrity verification using digital signatures. The TOE also supports the use of NTP or its own manually configurable time source to support the execution of reliable and time sensitive operations.
- TOE Access. The TOE can be accessed physically via direct serial connection or remotely via SSH connection. When a user account has reached the configured number of failed authentication attempts, the account will be locked for a Security Administrator defined time period.
- Trusted Path/Channels. The TOE protects the integrity and confidentiality of communications between itself and local/remote administrators.

2.2 TOE Models

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

Table 5: TOE Models

Model	Port Configuration	MACsec FPGA	CPU	Software
EX4400-24X	24-Port 1/10GbE SFP+	PHY1 BCM82756 PHY2 BCM82399 PHY3 BCM82398	Intel Atom C3558R (Denverton)	Junos OS 24.2R2
EX4400-24T	24-port 10/100/1000BASE-T	PHY1 BCM54192 PHY2 BCM82399		
EX4400-24P	24-port 10/100/1000BASE-T (PoE)	PHY1 BCM54192 PHY2 BCM82399		
EX4400-24MP	24x-port 100M/1/2.5/5/10GbE (PoE)	PHY1 BCM84894M PHY2 BCM82399		
EX4400-48T	48-port 10/100/1000BASE-T	PHY1 BCM54192 PHY2 BCM82399		
EX4400-48P	48-port 10/100/1000BASE-T (PoE)	PHY1 BCM54192 PHY2 BCM82399		
EX4400-48F	12-port 1000/10000BASE-X + 36-port 100/1000BASE-X	PHY1 BCM54195 PHY2 BCM82399 PHY3 BCM82756		
EX4400-48MP	12x100M/1/2.5/5/10GbE +	PHY1		

Model	Port Configuration	MACsec FPGA	CPU	Software
	36x100M/1/2.5GbE (PoE)	BCM54998EM PHY2 BCM82399 PHY3 BCM84894M		

2.2.1 Test Platform Equivalency

6 In summary, the evaluation team performed full testing on the EX4400-24P model. All other models are considered equivalent to the tested models. Detailed equivalency rationale is provided in Detailed Test Report.

3 Evaluation Activities for Mandatory SFRs

3.1 Security Audit (FAU)

3.1.1 FAU_GEN.1 Audit Data Generation

3.1.1.1 TSS

7 For the administrative task of generating/import of, changing, or deleting of cryptographic keys as defined in FAU_GEN.1.1c, the TSS shall identify what information is logged to identify the relevant key.

Findings
PASS
[ST] section 6.1.1 describes that the TOE logs the actions (generating, importing, changing or deleting) and key reference (Name or process ID) for cryptographic keys.

8 For distributed TOEs the evaluator shall examine the TSS to ensure that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components. The evaluator shall ensure that this mapping of audit events to TOE components accounts for, and is consistent with, information provided in Table 1, as well as events in Tables 2, 4, and 5 (where applicable to the overall TOE). This includes that the evaluator shall confirm that all components defined as generating audit information for a particular SFR should also contribute to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component cover all the SFRs that it implements.

EA Not Applicable	The TOE is not distributed.

3.1.1.2 Guidance Documentation

9 The evaluator shall check the guidance documentation and ensure that it provides an example of each auditable event required by FAU_GEN.1 (i.e. at least one instance of each auditable event, comprising the mandatory, optional and selection-based SFR sections as applicable, shall be provided from the actual audit record).

Findings
PASS
[AGD] section “Configuring Audit Log Options” contains an example of each auditable event required by FAU_GEN.1.

10 The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide

are related to TSF data related to configuration changes. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

Findings
PASS
The evaluator performed this activity as part of those AAs associated with ensuring the corresponding guidance documentation satisfied their independent requirements. However, overall, the evaluator considered the administrator guides published by the vendor. The evaluator reviewed the contents of the documentation and looked specifically for functionality related to the scope of the evaluation. Where there were missing or incomplete descriptions for the functionality such that the user could not complete the testing AAs, the evaluator requested the vendor to supply augmented guidance information. In the end, the vendor provided a more comprehensive guidance "supplement" document in the form of [AGD].

3.1.1.3 Tests

11 The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different identity and authentication (I&A) mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the guidance documentation, and that the fields in each audit record have the proper entries.

High-Level Test Description
Ensure each of the Mandatory, Optional, and Selection-based auditable requirements are met by reviewing the audit records generated during testing of the associated SFRs. Verify the audit records match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.
Findings
PASS

12 For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of auditable events to TOE components in the Security Target. For all events involving more than one TOE component when an audit event is triggered, the evaluator has to check that the event has been audited on both sides (e.g. failure of building up a secure communication channel between the two components). This is not limited to error cases but includes also events about successful actions like successful build up/tear down of a secure communication channel between TOE components.

Test Not Applicable	The TOE is not distributed.

13 Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.

3.1.2 FAU_GEN.2 User Identity Association

3.1.2.1 TSS & Guidance Documentation

14 The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

3.1.2.2 Tests

15 This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

16 For distributed TOEs the evaluator shall verify that where auditable events are instigated by another component, the component that records the event associates the event with the identity of the instigator. The evaluator shall perform at least one test on one component where another component instigates an auditable event. The evaluator shall verify that the event is recorded by the component as expected and the event is associated with the instigating component. It is assumed that an event instigated by another component can at least be generated for building up a secure channel between two TOE components. If for some reason (could be e.g. TSS or Guidance Documentation) the evaluator would come to the conclusion that the overall TOE does not generate any events instigated by other components, then this requirement shall be omitted.

Test Not Applicable	The TOE is not distributed.

3.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

3.1.3.1 TSS

17 The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.

Findings
PASS
[ST] section 6.1.1 states that the TOE sends audit data to an external syslog server via Netconf over SSH.

18 The evaluator shall examine the TSS to ensure it describes whether the TOE is a standalone TOE that stores audit data locally or a distributed TOE that stores audit data locally on each TOE component or a distributed TOE that contains TOE components that cannot store audit data locally on themselves but need to transfer audit data to other TOE components that can store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs it contains a list of TOE components that store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs that contain components which do not store audit data locally but transmit their generated audit data to other components it contains a mapping between the transmitting and storing TOE components.

Findings
PASS
[ST] section 6.1.1 states that the TOE is a standalone TOE that stores audit data locally.

19 The evaluator shall examine the TSS to ensure that it details whether the transmission of audit data to an external IT entity can be done in real-time, periodically, or both. In the case where the TOE is capable of performing transmission periodically, the evaluator shall verify that the TSS provides details about what event stimulates the transmission to be made as well as the possible acceptable frequency for the transfer of audit data.

Findings
PASS
[ST] section 6.1.1 states that the TOE sends the audit data to the external syslog server in real-time.

20 For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how audit data transfer to the external audit server is implemented among the different TOE components (e.g. every TOE components does its own transfer or the data is sent to another TOE component for central transfer of all audit events to the external audit server).

EA Not Applicable	The TOE is not distributed.

21 The evaluator shall examine the TSS to ensure it describes the amount of audit data that can be stored locally and how these records are protected against unauthorized modification or deletion.

Findings
PASS
[ST] section 6.1.1 states that the local storage for audit data on the TOE has a default maximum size of 1Gb and that only the Security Administrator can read or delete logs.

22 The evaluator shall examine the TSS to ensure it describes the method implemented for local logging, including format (e.g. buffer, log file, database) and whether the logs are persistent or non-persistent.

Findings
PASS
[ST] section 6.1.1 describes that the local logs are stored in a file on the TOE's storage volume.

23 The evaluator shall examine the TSS to ensure it describes the conditions that must be met for authorized deletion of audit records.

Findings
PASS
[ST] section 6.1.1, Only a Security Administrator can read or delete logs and archive files through the CLI interface or through direct access to the filesystem.

24 The evaluator shall examine the TSS to ensure it details the behaviour of the TOE when the storage space for audit data is full. When the option 'overwrite previous audit record' is selected this description should include an outline of the rule for overwriting audit data. If 'other actions' are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.

Findings
PASS
[ST] section 6.1.1 states, “When the maximum number of archive files is reached and when the size of the active file reaches the configured maximum size, the contents of the oldest archived file are deleted so the current active file can be archived.”

25 For distributed TOEs the evaluator shall examine the TSS to ensure it describes which TOE components are storing audit information locally and which components are buffering audit information and forwarding the information to another TOE component for local storage. For every component the TSS shall describe the behaviour when local storage space or buffer space is exhausted.

EA Not Applicable	The TOE is not distributed.

3.1.3.2 Guidance Documentation

26 The evaluator shall also examine the guidance documentation to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.

Findings
PASS
[AGD] section “Configuring Event Logging to a Remote Server when Initiating the Connection from the Remote Server” describes how to establish the trusted channel to the audit server, as well as any requirements on the audit server and configuration of the TOE needed to communicate with the audit server.

27 The evaluator shall also examine the guidance documentation to ensure it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and “cleared” periodically by sending the data to the audit server.

Findings
PASS
[AGD] section “Syslog Server Configuration on a Linux System” describes that the TOE stores the audit logs in a local audit file and simultaneously sends the logs to an external syslog server.

28 The evaluator shall examine the guidance documentation to ensure it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion.

Findings
PASS
There is no configuration required for protection of locally stored audit data against unauthorized modification or deletion. [ST] section 6.1.1 states, only a Security Administrator can read or delete logs and archive files through the CLI interface or through direct access to the filesystem.

29 If the storage size is configurable, the evaluator shall review the Guidance Documentation to ensure it contains instructions on specifying the required parameters.

Findings
PASS
[AGD] section “Configuring Audit Log Options in the Evaluated Configuration” contains instructions on specifying the required storage size parameters.

30 If more than one selection is made for FAU_STG_EXT.1.5, the evaluator shall review the Guidance Documentation to ensure it contains instructions on specifying which action is performed when the local storage space is full.

EA Not Applicable	More than one selection is not made for FAU_STG_EXT.1.5.

3.1.3.3 Tests

31 Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s).

32 The evaluator shall perform the following additional test for this requirement:

a. Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator’s choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing. The evaluator shall verify that the TOE is capable of transferring audit data to an external audit server automatically without administrator intervention.

High-Level Test Description
Establish a session between the TOE and the audit server. Confirm that the audit data is not able to be viewed in the clear and that the TOE is capable of transferring the audit data to an external audit server automatically without administrator intervention.
Findings
PASS - The evaluator confirmed the TOE is capable of transferring audit data to an external audit server automatically without administrator intervention. The logging server is OpenSSH v7.9p1 with bash 5.0.4 as described in the Test Setup.

b. Test 2: For distributed TOEs, Test 1 defined above shall be applicable to all TOE components that forward audit data to an external audit server.

Test Not Applicable	The TOE is not distributed.

c. Test 3: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall then make note of whether the TSS claims persistent or non-persistent logging and perform one of the following actions:

i. If persistent logging is selected, the evaluator shall perform a power cycle of the TOE and ensure that following power on operations the log events generated are still maintained within the local audit storage.

ii. If non-persistent logging is selected, the evaluator shall perform a power cycle of the TOE and ensure that following power on operations the log events generated are no longer present within the local audit storage.

High-Level Test Description
Generate audit data and verify that the TOE stores audit data locally. Reboot the TOE and confirm that the TOE's persistent storage retains the audit events.
Findings
PASS – The evaluator confirmed the log events generated are still maintained within the local audit storage following a power cycle.

d. Test 4: The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.5. Depending on the configuration this means that the evaluator shall check the content of the audit data when the audit data is just filled to the maximum and then verifies that:

i. The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option 'drop new audit data' in FAU_STG_EXT.1.5).

ii. The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option 'overwrite previous audit records' in FAU_STG_EXT.1.5)

iii. The TOE behaves as specified (for the option 'other action' in FAU_STG_EXT.1.5).

High-Level Test Description
Verify the oldest log file is overwritten when the configured number of archive files and the size is reached.
Findings
PASS – The evaluator confirmed the TOE supports overwriting the oldest audit records.

e. Test 5: For distributed TOEs, for the local storage according to FAU_STG_EXT.1.4, Test 1 specified above shall be applied to all TOE components that store audit data locally. For all TOE components that store audit data locally and comply with specified above shall be applied. The evaluator shall verify that the transfer of audit data to an external audit server is implemented.

Test Not Applicable	The TOE is not distributed.

f. Test 6 [Conditional]: In case manual export or ability to view locally is selected in FAU_STG_EXT.1.6, during interruption the evaluator shall perform a TSF-mediated action and verify the event is recorded in the audit trail.

NIAP TD0886

Note: The intent of the test is to ensure that the local audit TSF (as specified by FAU_STG_EXT.1.3) operates independently from the ability to transmit the generated audit data to an external audit server (as specified in FAU_STG_EXT.1.1). There are no specific requirements on the interruption of the connection between the TOE and the external audit server (as for FTP_ITC.1).

High-Level Test Description
Disable the connection to the remote audit server and perform a TSF-mediated action to confirm that the event is recorded in the local audit trail.
Findings
PASS - The administrator verified that the TSF-mediated event was recorded in the local audit trail.

3.2 Cryptographic Support (FCS)

3.2.1 NIAP Policy 5

33 To demonstrate that all cryptographic requirements are satisfied, the Assurance Activity Report must clearly indicate all SFRs for which a CAVP certificate is claimed and include, at a minimum, the cryptographic operation, the NIST standard, the SFR supported, the CAVP algorithm list name (e.g. AES, KAS, CVL, etc.) and the CAVP Certificate number.

SFR	Cryptographic Operation	NIST Standard	CAVP Certificate (Algorithm)
FCS_CKM.1	Asymmetric Key Generation: 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;	FIPS PUB 186-4 FIPS PUB 186-5	A6931 (RSA)
FCS_CKM.1	Asymmetric Key Generation: 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;	FIPS PUB 186-4	A6931 (ECDSA)
FCS_CKM.2	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;	NIST SP 800-56A Revision 3	A6931 (KAS-ECC-SCC Sp800-56Ar3)
FCS_COP.1/ DataEncryption	encryption/decryption in accordance with a specified cryptographic algorithm AES used in [CBC, CTR] mode and	FIPS PUB 197	A6931 (AES)

SFR	Cryptographic Operation	NIST Standard	CAVP Certificate (Algorithm)
	cryptographic key sizes [128 bits, 256 bits]	NIST SP 800-38A	
FCS_COP.1/ SigGen	cryptographic signature services (generation and verification): RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits, 3072-bits, 4096 bits]	FIPS PUB 186-4 FIPS PUB 186-5	A6931 (RSA)
FCS_COP.1/ SigGen	cryptographic signature services (generation and verification): Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits, 384 bits, 521 bits]	FIPS PUB 186-4	A6931 (ECDSA)
FCS_COP.1/ Hash	cryptographic hashing services in accordance with a specified cryptographic algorithm [SHA-256, SHA-384, SHA-512]	FIPS PUB 180-4	A6931 (SHA-256, SHA-384, SHA-512)
FCS_COP.1/ Hash	cryptographic hashing services in accordance with a specified cryptographic algorithm [SHA-256, SHA-512]	FIPS PUB 180-4	A6930 (SHA-256, SHA-512)
FCS_COP.1/ Hash	cryptographic hashing services in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512]	FIPS PUB 180-4	A6929 (SHA-1, SHA-256, SHA-384, SHA-512)
FCS_COP.1/ KeyedHash	keyed-hash message authentication in accordance with a specified cryptographic algorithm [HMAC-SHA-256, HMAC-SHA-512, implicit]	FIPS PUB 198-1	A6931 (HMAC-SHA-256, HMAC-SHA-512)
FCS_COP.1/MAC SEC	[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].	NIST SP 800-38F	A6928 (AES KeyWrap)
FCS_COP.1/MAC SEC	[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	FIPS PUB 197 NIST SP 800-38D	AES 4544(AES GCM) AES 4545 (AES GCM)

SFR	Cryptographic Operation	NIST Standard	CAVP Certificate (Algorithm)
	Wrap as specified in NIST SP 800- 38F, GCM as specified in ISO 19772].		AES 4550(AES GCM) C996 (AES GCM) C1869 (AES GCM)
FCS_COP.1/CMAC	[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].	NIST SP 800-38B	A6928 (AES-CMAC)
FCS_RBG_EXT.1	deterministic random bit generation services using [HMAC_DRBG [SHA-512]]	NIST SP 800-90A	A6929 (HMAC)

3.2.2 FCS_CKM.1 Cryptographic Key Generation

3.2.2.1 TSS

34 The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

Findings

PASS

[ST] section 6.2.1 identifies all key sizes supported by the TOE and the usage for each scheme.

3.2.2.2 Guidance Documentation

35 The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target.

Findings

PASS

[AGD] section “Critical Security Parameters” describes that the RSA and ECDSA keys are generated the first time SSH is configured.

[AGD] section “Configuring SSH on the Evaluated Configuration” instructs the administrator how to configure the TOE to use the selected key generation schemes and key sizes for all cryptographic protocols defined in the security target.

3.2.2.3 Tests

NIAP TD0921

36 Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported).

Key Generation for FIPS PUB 186-4 or FIPS PUB 186-5 RSA Schemes

37 The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d. This test must be repeated for each supported RSA modulo and generation method.

38 Key Pair generation specifies 5 ways (or methods) to generate the primes *p* and *q*. These include:

a. Random Provable Primes (p and q shall be provable primes)

b. Random Probable primes (p and q shall be probable primes)

c. Provable Primes with Conditions (p1, p2, q1, q2, p and q shall all be provable primes)

d. Provable/probable primes with Conditions (p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes)

e. Probable primes with Conditions (p1, p2, q1, q2, p and q shall all be probable primes)

39 The Random Provable primes, and all the Primes with Conditions can be tested in the same manner because each of these begin with a starting random number and calculate the p and q values from this value. The test instructs the TSF to generate intermediate values and the p, q, n, and d values. The evaluator then validates the correctness of the values generated by the TSF.

40 To test the key generation method for the Random Provable primes method or Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. If the TSF provides the input to the key generation function, such input must be recorded and verified. For each RSA key length (modulo) claimed, the evaluator shall generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing Key Pair values generated by the TSF with those generated using a same set of input values using a known good implementation.

41 The Random Probable primes must be tested in a different way because this test generates different random numbers, not related to each other, until the number satisfies the "probably prime" requirements. This validation method requires two tests for Random Probable

primes. These include the Known Answer Test and the Miller-Rabin probabilistic primality test.

42 To test the key generation method for the Random Probable primes, the known answer test must be used. The evaluator shall compare the results of a known good implementation with the TSF results for a set (of a corresponding size depending on modulus) of supplied values containing private prime factor p, private prime factor q and confirm all public keys, e, match. Then the evaluator shall use the TSF to generate prime p, q pairs for each modulus size and perform the Miller-Rabin tests.

Note	[ST] Table 16 specifies the CAVP certificate demonstrating the TOE correctly implements RSA key generation.

Key Generation for Elliptic Curve Cryptography (ECC)

Key Generation for ECDSA Schemes

43 Key pairs for the ECDSA consist of pairs (d, Q), where the private key, d, is an integer, and the public key, Q, is an elliptic curve point.

44 The evaluator shall verify the implementation of ECC Key Generation by the TOE using the following components tests:

- ECC Key Pair Generation
- ECC Public Key Validation (PKV)

ECC Key Pair Generation Test

45 There are four methods by which these pairs may be generated:

- FIPS 186-4 Section B.4.1 Key Pair Generation Using Extra Random Bits

  Using this method, 64 more bits are requested from the RBG than are needed for d so that bias produced by the mod function is negligible.

- FIPS 186-4 Section B.4.2 Key Pair Generation by Testing Candidates

  Using this method, a random number is obtained and tested to determine that it will produce a value of d in the correct range. If d is out-of-range, another random number is obtained (i.e., the process is iterated until an acceptable value of d is obtained.

- FIPS 186-5 Section A.2.1 ECDSA Key Pair Generation using Extra Random Bits

  Using this method, more bits are requested from the DRBG than are needed for d so that the bias produced by the mod function is negligible.

- FIPS 186-5 Section A2.2 ECDSA Key Pair Generation by Rejection Sampling

  Using this method, a random number is obtained and tested to determine that it will produce a value of d in the correct range. If d is out of range, an ERROR is returned.

46 The evaluator shall test the ECC Key Pair Generation by having the TSF produce 10 key pairs (d, Q) for each implemented key generation method using each supported curve (i.e., P-256, P-384, and P-521). The steps are the same for each key generation method and curve. The private key, d, shall be generated using the output of an approved DRBG converted to an integer via modular reduction or the discard method. The known private key is then used by the TSF to compute the public key, Q'. To evaluator then validates the correctness by comparing the value Q' computed by the TSF to the public key, Q generated by a known good implementation.

ECC Public Key Verification (PKV) Test

47 The evaluator shall generate 12 key pairs (d, Q) for each selected curve, with 6 valid public keys, Q, using a known good implementation and 6 modified, Q', and determine whether the TSF can accurately detect these modifications. Q' should be otherwise valid but include at least one of the following errors: a) point X or Y not on the curve, b) point X or Y is too large for the field for the given curve. The evaluator encouraged to make sure that the modification does not inadvertently result in another valid public key (e.g., modifying Y and accidentally hitting a different point on the curve).

Note	[ST] Table 16 specifies the CAVP certificate demonstrating the TOE correctly implements ECDSA key generation.

Key Generation for Finite-Field Cryptography (FFC)

48 The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the TOE using the Parameter Generation and Key Generation test. This test verifies the ability of the TSF to correctly produce values for the field prime p, the cryptographic prime q (dividing p-1), the cryptographic group generator g, and the calculation of the private key x and public key y.

49 The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p:

- Primes q and p shall both be provable primes
- Primes q and field prime p shall both be probable primes

50 and two ways to generate the cryptographic group generator g:

- Generator g constructed through a verifiable process
- Generator g constructed through an unverifiable process.

51 The Key generation specifies 2 ways to generate the private key x:

- len(q) bit output of RBG where $1 \Leftarrow x \Leftarrow q-1$
- len(q) + 64 bit output of RBG, followed by a mod q-1 operation and a +1 operation, where $1 \Leftarrow x \Leftarrow q-1$.

52 The security strength of the RBG must be at least that of the security offered by the FFC parameter set.

53 To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set.

54 For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm

- g != 0,1
- q divides p-1
- g^q mod p = 1
- g^x mod p = y

55 for each FFC parameter set and key pair.

FFC Schemes using "safe-prime"

56 Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1.

Note	This test case is covered in FCS_CKM.2 testing.

3.2.3 FCS_CKM.2 Cryptographic Key Establishment

3.2.3.1 TSS

57 The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme. It is sufficient to provide the scheme, SFR, and service in the TSS.

58 The intent of this activity is to be able to identify the scheme being used by each service. This would mean, for example, one way to document scheme usage could be as shown in the table below. The information provided in this example does not necessarily have to be included as a table but can be presented in other ways as long as the necessary data is available.

Scheme	SFR	Service
RSA	FCS_TLSS_EXT.1	Administration
ECDH	FCS_IPSEC_EXT.1	Authentication Server

Findings
PASS
[ST] section 6.2.1 identifies all supported key establishment schemes and the usage for each scheme.

3.2.3.2 Guidance Documentation

59 The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” describes how to configure the TOE to use the selected key establishment scheme.

3.2.3.3 Tests

Key Establishment Schemes

60 The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE using the applicable tests below.

ECC and FIPS 186-type FFC SP800-56A Key Establishment Schemes

61 The evaluator shall verify a TOE’s implementation of SP800-56A key agreement schemes using the following Function and Validity tests for ECC and FIPS186- type. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag.

Function Test

62 The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested.

63 The evaluator shall obtain the DKM, the corresponding TOE’s public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields.

64 If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret.

65 The evaluator shall verify the correctness of the TSF’s implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values.

66 If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm.

Validity Test

67 The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOE's public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE id fields.

68 The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator shall also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOE's static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass).

69 The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results with the results using a known good implementation verifying that the TOE detects these errors.

Note	[ST] Table 16 specifies the CAVP certificate demonstrating the TOE correctly implements ECC SP 800-56A key agreement/establishment schemes.

RSA-based key establishment schemes

70 The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses RSAES-PKCS1-v1_5.

Test Not Applicable	The [ST] does not select RSA-based key establishment schemes.

FFC Schemes using "safe-prime" groups

71 The evaluator shall verify the correctness of the TSF's implementation of safe-prime groups by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses safe-prime groups. This test must be performed for each safe-prime group that each protocol uses.

Test Not Applicable	The ST does not claim the use of safe-prime key establishment.

3.2.4 FCS_CKM.4 Cryptographic Key Destruction

3.2.4.1 TSS

72 The evaluator shall examine the TSS to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations (e.g. factory reset or device wipe function, disconnection of trusted channels, key change as part of a secure channel protocol), and the destruction method used in each case. For the purpose of this Evaluation Activity the relevant keys are those keys that are relied upon to support any of the SFRs in the Security Target. The evaluator shall confirm that the description of keys and storage locations is consistent with the functions carried out by the TOE (e.g. that all keys for the TOE-specific secure channels and protocols, or that support FPT_APW.EXT.1 and FPT_SKP_EXT.1, are accounted for[2]). In particular, if a TOE claims not to store plaintext keys in non-volatile memory then the evaluator shall check that this is consistent with the operation of the TOE.

Findings
PASS
[ST] Section 6.2.1 lists all relevant keys, key destruction situations and the destruction method used in each case. The evaluator confirms that the description of keys and storage locations is consistent with the functions carried out by the TOE.

73 The evaluator shall check to ensure the TSS identifies how the TOE destroys keys stored as plaintext in non-volatile memory, and that the description includes identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).

Findings
PASS
[ST] section 6.2.1 identifies how the TOE destroys keys stored as plaintext in non-volatile memory through the ‘request system zeroize’ option.

74 Note that where selections involve ‘*destruction of reference*’ (for volatile memory) or ‘*invocation of an interface’* (for non-volatile memory) then the relevant interface definition is examined by the evaluator to ensure that the interface supports the selection(s) and description in the TSS. In the case of non-volatile memory the evaluator includes in their examination the relevant interface description for each media type on which plaintext keys are stored. The presence of OS-level and storage device-level swap and cache files is not examined in the current version of the Evaluation Activity.

Findings
PASS
[ST] section 6.2.1 describes that the TOE software calls the free() function when the key is no longer needed.

75 Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator shall check that the TSS identifies the encryption method and the key-encrypting-key used, and that the key-encrypting-key is either itself stored in an encrypted form or that it is destroyed by a method included under FCS_CKM.4.

Findings
PASS
[ST] section 6.2.1 describes that the “SHA-512 is implemented in the LibMD library and used for password hashing”.

76 The evaluator shall check that the TSS identifies any configurations or circumstances that may not conform to the key destruction requirement (see further discussion in the Guidance Documentation section below). Note that reference may be made to the Guidance Documentation for description of the detail of such cases where destruction may be prevented or delayed.

Findings
PASS
[ST] section 6.2.1 The TOE does not have any circumstances that may not conform to key destruction requirements.

77 Where the ST specifies the use of “a value that does not contain any CSP” to overwrite keys, the evaluator shall examine the TSS to ensure that it describes how that pattern is obtained and used, and that this justifies the claim that the pattern does not contain any CSPs.

Note	The ST does not specify “a value that does not contain any CSP.”

3.2.4.2 Guidance Documentation

78 A TOE may be subject to situations that could prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS (and any other supporting information used). The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.

79 For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-levelling and garbage collection. This may result in additional copies of the key that are logically inaccessible but persist physically. Where available, the TOE might then describe use of the TRIM command[1] and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

Findings
PASS
There are no circumstances where delayed or prevented key destruction can occur. The “Zeroizing the System” section in the [AGD] describes the process for clearing CSPs and other sensitive information from the TOE when required.

[1] Where TRIM is used then the TSS and/or guidance documentation is also expected to describe how the keys are stored such that they are not inaccessible to TRIM, (e.g. they would need not to be contained in a file less than 982 bytes which would be completely contained in the master file table).

3.2.4.3 Tests

80 None

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

3.2.5.1 TSS

81 The evaluator shall examine the TSS to ensure it identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption.

Findings
PASS
[ST] section 6.2.1 identifies that the TOE performs data encryption/decryption using AES-CBC and AES-CTR modes with 128- and 256-bit keys.

3.2.5.2 Guidance Documentation

82 The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected mode(s) and key size(s) defined in the Security Target supported by the TOE for data encryption/decryption.

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” instructs the administrator how to configure the TOE to use the selected modes and key sizes supported by the TOE for data encryption/decryption.

3.2.5.3 Tests

AES-CBC Known Answer Tests

83 There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

84 **KAT-1.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key.

85 To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.

86 **KAT-2.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-

zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys.

87 To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.

88 **KAT-3.** To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key *i* in each set shall have the leftmost *i* bits be ones and the rightmost *N-i* bits be zeros, for *i* in [1,N].

89 To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key.

90 **KAT-4.** To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].

91 To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.

AES-CBC Multi-Block Message Test

92 The evaluator shall test the encrypt functionality by encrypting an *i*-block message where 1 < *i* <=10. The evaluator shall choose a key, an IV and plaintext message of length *i* blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.

93 The evaluator shall also test the decrypt functionality for each mode by decrypting an *i*-block message where 1 < *i* <=10. The evaluator shall choose a key, an IV and a ciphertext message of length *i* blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.

AES-CBC Monte Carlo Tests

94 The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

Input: PT, IV, Key

```
for i = 1 to 1000:
    if i == 1:
        CT[1] = AES-CBC-Encrypt(Key, IV, PT)
        PT = IV
    else:
        CT[i] = AES-CBC-Encrypt(Key, PT)
        PT = CT[i-1]
```

95 The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

96 The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AES-CBC-Decrypt.

AES-GCM Test

97 The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:

128 bit and 256 bit keys

1 **Two plaintext lengths**. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.

a) **Three AAD lengths**. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.

b) **Two IV lengths**. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.

98 The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.

99 The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.

100 The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

AES-CTR Known Answer Tests

101 The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. Since the Counter Mode does not specify the counter that is used, it is not possible to implement an automated test for this mode. The generation and management of the counter is tested if the TSF is validated against the requirements of the Functional Package for Secure Shell referenced in Section 2.2 of the cPP. If CBC and/or GCM are selected in FCS_COP.1/DataEncryption, the test activities for those modes sufficiently demonstrate the correctness of the AES algorithm. If CTR is the only selection in FCS_COP.1/DataEncryption, the AES-CBC Known Answer Test, AES-GCM Known Answer Test, or the following test shall be performed (all of these tests demonstrate the correctness of the AES algorithm):

102 There are four Known Answer Tests (KATs) described below to test a basic AES encryption operation (AES-ECB mode). For all KATs, the plaintext, ~~IV,~~ and ciphertext values shall be 128-bit blocks. The results from each test may either be obtained by the validator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

103 KAT-1 To test the encrypt functionality, the evaluator shall supply a set of 5 plaintext values for each selected keysize and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros.

104 KAT-2 To test the encrypt functionality, the evaluator shall supply a set of 5 key values for each selected keysize and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value.

105 KAT-3 To test the encrypt functionality, the evaluator shall supply a set of key values for each selected keysize as described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values. A set of 128 128-bit keys, a set of 192 192-bit keys, and/or a set of 256 256-bit keys. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N].

106 KAT-4 To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the ciphertext values that result from encryption of the given plaintext using each selected keysize with a key value of all zeros (e.g. 256 ciphertext values will be generated if 128 bits and 256 bits are selected and 384 ciphertext values will be generated if all keysizes are selected). Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128]

AES-CTR Multi-Block Message Test

107 The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10 (test shall be performed using AES-ECB mode). For each i the evaluator shall choose a key and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key using a known good implementation. The evaluator shall perform this test using each selected keysize.

AES-CTR Monte-Carlo Test

108 The evaluator shall test the encrypt functionality using 100 plaintext/key pairs. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows:

Input: PT, Key

for i = 1 to 1000:

CT[i] = AES-ECB-Encrypt(Key, PT) PT = CT[i]

109 The ciphertext computed in the 1000th iteration is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation. The evaluator shall perform this test using each selected keysize.

110 There is no need to test the decryption engine.

Note	[ST] Table 16 specifies the CAVP certificate demonstrating the TOE correctly implements AES-CBC and AES-CTR. [ST] Table 20 specifies the CAVP certificate demonstrating the TOE correctly implements AES-GCM. This is required by FCS_COP.1/MACSEC.

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

3.2.6.1 TSS

111 The evaluator shall examine the TSS to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services.

Findings
PASS
[ST] section 6.2.1 describes that the TOE supports RSA with 2048-, 3072-and 4096-bit keys or ECDSA with 256, 384 and 521 bit keys for performing signature services.

3.2.6.2 Guidance Documentation

112 The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected cryptographic algorithm and key size defined in the Security Target supported by the TOE for signature services.

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” instructs the administrator how to configure the TOE to use the selected cryptographic algorithm and key size for signature services.

3.2.6.3 Tests

NIAP TD0921

ECDSA Signature Algorithm Tests

113 The evaluator shall verify the implementation of ECDSA Signature generation by the TOE using the Signature Generation Test. This test validates the TSF generation of the (r, s) pair that represent the digital signature. The digital signature shall be verified using the same domain parameters and hash function that were used during signature generation. An approved hash function or an XOF shall be used for this purpose.

Signature Generation Test

114 The purpose of this test is to verify the ability of the TSF to produce correct signatures.

115 To test signature generation, the evaluator supplies 10 pseudorandom messages to the TSF and a key pair, (d, Q), generated by a known good implementation. Exercising each applicable curve (i.e., P-256, P 384, or P-521) and hash algorithm or extendable-output function combination, the TSF generates signatures for each supplied message and returns the corresponding signatures. Using a known-good implementation, the evaluator validates the signatures by using the associated public key, Q, to verify the signature.

Signature Verification Test

116 The purpose of this test is to verify the ability of the TSF to accept valid signatures and reject invalid signatures.

117 For each curve/hash algorithm or extendable-output function combination supported by the TSF, the evaluator shall use a known good implementation to generate a key pair, (d, Q), and use known good implementation with the private key, d, to sign 15 pseudorandom messages of 1024 bits. The evaluator shall alter some of the messages or signatures so that signature verification should fail.

118 To test signature verification, the evaluator supplies the public key, Q, and 15 pseudorandom messages, including altered messaged, to the TSF for verification of signatures. The evaluator shall then verify that the TSF validates correct signatures on the original messages and flags or rejects the altered messages.

RSA Signature Algorithm Tests

Signature Generation Test

119 The evaluator shall verify the implementation of RSA Signature generation by the TOE using the Signature Generation Test. This test verifies the ability of the TSF to produce correct signatures.

120 There are 2 different RSA Signature algorithms that can be implemented. These include:

a. RSASSA-PKCS1-v1.5

b. RSASSA-PSS

121 To test signature generation, the evaluator generates or obtains 10 messages for each modulus size/hash or extendable-output function combination supported by the TOE. Using

a key generated by a known good implementation, the TSF generates and returns the corresponding signatures to a known good implementation that validates the signatures by using the associated public key to verify the signature.

122 The evaluator shall verify the correctness of the TOE's signature using a trusted reference implementation of the signature verification algorithm and the associated public keys to verify the signatures.

Signature Verification Test

123 The evaluator shall verify the implementation of RSA Signature verification by the TOE using the Signature Verification Test. This test verifies the ability of the TSF to recognize valid and invalid signatures.

124 There are 2 different RSA Signature algorithms that can be implemented. These include:

a. RSASSA-PKCS1-v1.5

b. RSASSA-PSS

125 For each modulus size/hash or extendable-output function combination supported by the TOE, the evaluator shall use a known good implementation to generate a modulus and three associated key pairs, (d, e). Each private key d is used to sign six pseudorandom messages each of 1024 bits. Some of the public keys, e, messages, IR format, or signatures must be altered so that signature verification should fail. The modifications must cover distinct "key modified", "message modified", "signature modified", "IR moved", and "trailer moved" tests. The modulus, hash or extendable-output algorithm, public key e values, messages, and signatures are forwarded to the TSF. The TSF then attempts to verify the signatures and returns the results. The evaluator then compares the received results with the stored results from a known good implementation.

126 The evaluator shall verify that the TSF validates correct signatures on the original messages and flags or rejects the altered messages.

Note	[ST] Table 16 specifies the CAVP certificate demonstrating the TOE correctly implements RSA and ECDSA signature generation and verification.

3.2.7 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

3.2.7.1 TSS

127 The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS.

Findings
PASS
[ST] section 6.2.1 describes the association of the hash function with other TSF cryptographic functions.

3.2.7.2 Guidance Documentation

128 The evaluator shall check the AGD documents to determine that any configuration that is required to configure the required hash sizes is present.

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” describes how to configure the required hash sizes used in message authentication algorithms.
[AGD] section “Configuring a Network Device collaborative Protection Profile for an Authorized Administrator” describes how to configure the hash algorithm and sizes used for password storage.

3.2.7.3 Tests

129 The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented testmacs.

130 The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP.

Short Messages Test - Bit-oriented Mode

131 The evaluator shall devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluator shall compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Short Messages Test - Byte-oriented Mode

132 The evaluator shall devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test - Bit-oriented Mode

133 The evaluator shall devise an input set consisting of m messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test - Byte-oriented Mode

134 The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m +

8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Pseudorandomly Generated Messages Test

135 This test is for byteoriented implementations only. The evaluator shall randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluator shall then ensure that the correct result is produced when the messages are provided to the TSF.

Note	[ST] Table 16 specifies the CAVP certificates demonstrating the TOE correctly implements Hashing.

3.2.8 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

3.2.8.1 TSS

136 The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

Findings
PASS
[ST] section 6.2.1 describes the key length, hash functions used, block sizes and output MAC lengths used for the HMAC functions.

3.2.8.2 Guidance Documentation

137 The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the values used by the HMAC function: key length, hash function used, block size, and output MAC length used defined in the Security Target supported by the TOE for keyed hash function.

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” describes how to configure the TOE to use the values used by the HMAC functions supported by the TOE for keyed hash functions.

3.2.8.3 Tests

138 For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and message data using a known good implementation.

Note	[ST] Table 16 specifies the CAVP certificate demonstrating the TOE correctly implements HMAC algorithms.

3.2.9 FCS_RBG_EXT.1: Cryptographic Operation (Random Bit Generation)

139 Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Appendix D of [NDcPP].

3.2.9.1 TSS

140 The evaluator shall examine the TSS to determine that it specifies the DRBG type, identifies the entropy source(s) seeding the DRBG, and state the assumed or calculated min-entropy supplied either separately by each source or the min-entropy contained in the combined seed value.

Findings
PASS
[ST] section 6.2.1 describes the HMAC_DRBG (SHA-512) seeded from a platform-based entropy source which generates 256 bits of entropy.

3.2.9.2 Guidance Documentation

141 The evaluator shall confirm that the guidance documentation contains appropriate instructions for configuring the RNG functionality.

Findings
PASS
[AGD] section “Enabling FIPS Mode” contains instructions for configuring the RNG functionality.

3.2.9.3 Tests

142 The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration.

143 If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator shall verify that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A).

144 If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator shall verify that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the

instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

145 The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

Entropy input: the length of the entropy input value must equal the seed length.

Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length.

Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.

Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

Note	[ST] Table 16 specifies the CAVP certificate demonstrating the TOE correctly implements Deterministic Random Bit Generation.

3.3 Identification and Authentication (FIA)

3.3.1 FIA_UIA_EXT.1 User Identification and Authentication

3.3.1.1 TSS

146 The evaluator shall examine the TSS to determine that it describes the logon process for remote authentication mechanism (e.g. SSH public key, Web GUI password, etc.) and optional local authentication mechanisms supported by the TOE. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a “successful logon”.

Findings
PASS
[ST] section 6.3.1 describes the logon process for local and remote authentication and what constitutes a “successful logon”. The TOE supports remote SSH logon with username and password or SSH PublicKeys, and local serial access with username and password.

147 The evaluator shall examine the TSS to determine that it describes which actions are allowed before administrator identification and authentication. The description shall cover authentication and identification for local and remote TOE administration

Findings
PASS
[ST] section 6.3.1 states “Prior to authentication, the TOE shall only allow displaying of an access banner, responding to an ICMP echo, and negotiation of a SSH session.”

148 For distributed TOEs the evaluator shall examine that the TSS details how Security Administrators are authenticated and identified by all TOE components. If not all TOE components support authentication of Security Administrators according to FIA_UIA_EXT.1, the TSS shall describe how the overall TOE functionality is split between TOE components including how it is ensured that no unauthorized access to any TOE component can occur.

EA Not Applicable	The TOE is not distributed.

149 For distributed TOEs, the evaluator shall examine the TSS to determine that it describes for each TOE component which actions are allowed before administrator identification and authentication. The description shall cover authentication and identification for remote TOE administration and optionally for local TOE administration if claimed by the ST author. For each TOE component that does not support authentication of Security Administrators according to FIA_UIA_EXT.1 the TSS shall describe any unauthenticated services/services that are supported by the component.

EA Not Applicable	The TOE is not distributed.

3.3.1.2 Guidance Documentation

150 The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the guidance documentation provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the guidance documentation provides sufficient instruction on limiting the allowed services.

Findings
PASS
[AGD] sections “Configuring Security Administrator and FIPS User Identification and Access”, “Configuring a Network Device collaborative Protection Profile for an Authorized Administrator” and “Configuring SSH on the Evaluated Configuration” describe all the preparatory steps necessary to log in. [AGD] section “Authentication Methods in FIPS Mode of Operation” provides clear instructions for successfully logging on for each supported login method.
There is no configuration necessary to ensure the services provided before login are limited.

3.3.1.3 Tests

151 The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

a. Test 1: The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For all combinations of supported credentials and login methods, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

High-Level Test Description
For each management interface and credential type: • Log into using a known-good credential and verify login succeeds. • Log into using a known-bad credential and verify login fails.
Findings
PASS – The evaluator confirmed correct username and password or public key results in the ability to access the system, while providing incorrect information results in denial of access.

b. Test 2: The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.

High-Level Test Description
Attempt to access the services identified in the SFR as available to external remote entities and confirm they are accessible. Run a port scan and confirm that no other unclaimed services are available.
Findings
PASS – The evaluator confirmed that the list of services available is limited to those specified in the requirement.

c. Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

High-Level Test Description
At the Local Console, enter common shell key combinations and strings to escape and/or run commands. Verify the user is unable to run any commands or services other than the warning banner.
Findings
PASS – The evaluator confirmed the TOE is unable to run any commands or services that are limited.

d. Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.

Test Not Applicable	The TOE is not distributed.

3.4 Security management (FMT)

3.4.1 General Requirements for Distributed TOEs

3.4.1.1 TSS

152 For distributed TOEs, the evaluator shall verify that the TSS describes how every function related to security management is realized for every TOE component and shared between

different TOE components. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

EA Not Applicable	The TOE is not distributed.

3.4.1.2 Guidance Documentation

153 For distributed TOEs, the evaluator shall verify that the Guidance Documentation to describe management of each TOE component. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

EA Not Applicable	The TOE is not distributed.

3.4.1.3 Tests

154 Tests defined to verify the correct implementation of security management functions shall be performed for every TOE component. For security management functions that are implemented centrally, sampling should be applied when defining the evaluator's tests (ensuring that all components are covered by the sample).

Test Not Applicable	The TOE is not distributed.

3.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

3.4.2.1 TSS

155 For distributed TOEs see [ND-SD] Section 2.4.1.1. There are no specific requirements for non-distributed TOEs.

Note	The TOE is not a distributed TOE.

3.4.2.2 Guidance Documentation

156 The evaluator shall examine the guidance documentation to determine that any necessary steps to perform manual update are described. The guidance documentation shall also provide warnings regarding functions that may cease to operate during the update (if applicable).

Findings
PASS
[AGD] section "Install Junos OS Software Package" describes the necessary steps to perform a manual update and describes that the TOE will need to reboot to load the new software.

157 For distributed TOEs the guidance documentation shall describe all steps how to update all TOE components. This shall contain description of the order in which components need to be updated if the order is relevant to the update process. The guidance documentation

shall also provide warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update (if applicable).

EA Not Applicable	The TOE is not distributed.

3.4.2.3 Tests

158 The evaluator shall perform the following tests:

a. Test 1: The evaluator shall try to perform the update using a legitimate update image without prior authentication as Security Administrator (either by authentication as a user with no administrator privileges or without user authentication at all – depending on the configuration of the TOE). The attempt to update the TOE shall fail.

High-Level Test Description
As a user without Security Administrator permissions attempt to initiate an update with a legitimate update image. Confirm that the update fails.
Findings
PASS – The evaluator confirmed the TOE is unable to perform an update as a user with no administrator privileges.

b. Test 2: The evaluator shall try to perform the update with prior authentication as Security Administrator using a legitimate update image. This attempt should be successful. This test case should be covered by the tests for FPT_TUD_EXT.1 already.

Note	This is covered by FPT_TUD_EXT.1 Test 1.

3.4.3 FMT_MTD.1/CoreData Management of TSF Data

3.4.3.1 TSS

159 For each administrative function identified in the guidance documentation that is accessible through an interface prior to administrator log-in, the evaluator shall confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

Findings
PASS
[ST] section 6.3.1 claims that only the access banner, responding to ICMP echo and negotiation of an SSH session are accessible prior to administrator log-in. The TOE requires users to successfully log-in before any controlled access is granted.

160 If the TOE supports handling of X.509v3 certificates and implements a trust store, the evaluator shall examine the TSS to determine that it contains sufficient information to describe how the ability to manage the TOE's trust store is restricted.

EA Not Applicable	The ST does not support handling of X.509v3 certificates.

3.4.3.2 Guidance Documentation

161 The evaluator shall review the guidance documentation to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the cPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

Findings
PASS
The [AGD] lists all the functions that can be used to manipulate TSF data. For specific references, please refer to the SFR AA of interest. For example, SSH protocol TSF data manipulating functions are described in FCS_SSH_EXT.1.*.

162 If the TOE supports handling of X.509v3 certificates and provides a trust store, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to configure and maintain the trust store in a secure way. If the TOE supports loading of CA certificates, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to securely load CA certificates into the trust store. The evaluator shall also review the guidance documentation to determine that it explains how to designate a CA certificate a trust anchor.

EA Not Applicable	The ST does not support handling of X.509v3 certificates.

3.4.3.3 Tests

163 No separate testing for FMT_MTD.1/CoreData is required unless one of the management functions has not already been exercised under any other SFR.

3.4.4 FMT_SMF.1 Specification of Management Functions

164 The security management functions for FMT_SMF.1 are distributed throughout the cPP and are included as part of the requirements in FTA_SSL_EXT.1, FTA_SSL.3, FTA_TAB.1, FMT_MOF.1/ManualUpdate, FMT_MOF.1/AutoUpdate (if included in the ST), FIA_AFL.1, FIA_X509_EXT.2.2 (if included in the ST), FPT_TUD_EXT.1.2 & FPT_TUD_EXT.2.2 (if included in the ST and if they include an administrator-configurable action), FMT_MOF.1/Services, and FMT_MOF.1/Functions (for all of these SFRs that are included in the ST), FMT_MTD, FPT_TST_EXT, and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FMT_SMF.1.

3.4.4.1 TSS (containing also requirements on Guidance Documentation and Tests)

165 The evaluator shall examine the TSS, Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE. The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).

Findings

PASS

The evaluator examined the [ST] section 6.4.1, the [AGD] throughout and the TOE during all other testing and confirmed that the management functions specified in FMT_SMF.1 are provided by the TOE. The administrator is able to securely manage the TOE remotely via SSH or through local serial interface. There are no restrictions on the availability of security management functions on a specific interface.

166 The evaluator shall examine the TSS and Guidance Documentation to verify they both describe the local administrative interface. The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

Findings

PASS

[ST] section 6.4.1 describes the local console as being connected locally via the serial ports on the TOE.

[AGD] section “Understanding Management Interfaces” describes the local administrative interface. The [AGD] describes the local interface as a RJ-45 console port on the rear of the device. This specifies that the interface is local.

167 For distributed TOEs with the option 'ability to configure the interaction between TOE components' the evaluator shall examine that the ways to configure the interaction between TOE components is detailed in the TSS and Guidance Documentation. The evaluator shall check that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and Guidance Documentation.

EA Not Applicable The TOE is not distributed.

168 (If configure local audit is selected) The evaluator shall examine the TSS and Guidance Documentation to ensure that a description of the logging implementation is described in enough detail to determine how log files are maintained on the TOE.

Findings

PASS

[ST] section 6.1.1 provides a description of the logging implementation in enough detail to determine how log files are maintained on the TOE. The TOE has an active log file of a maximum size of 1Gb and when the active file reaches its maximum size it is compressed and archived. The TOE keeps an administrator configured number of archived log files, and when active log is full and the maximum number of archived logs exists, the TOE will delete the oldest archived log first.

[AGD] section “Configuring Audit Log Options” provides instructions on configuring the audit logs on the TOE in enough detail to determine how log files are maintained.

3.4.4.2 Guidance Documentation

169 See [ND-SD] section 2.4.4.1.

3.4.4.3 Tests

170 The evaluator shall test management functions as part of testing the SFRs identified in [ND-SD] Section 2.4.4. No separate testing for FMT_SMF.1 is required unless one of the

management functions in FMT_SMF.1.1 has not already been exercised under any other SFR.

3.4.5 FMT_SMR.2 Restrictions on Security Roles

3.4.5.1 TSS

171 The evaluator shall examine the TSS to determine that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE (e.g. if local administrators and remote administrators have different privileges or if several types of administrators with different privileges are supported by the TOE).

Findings
PASS
[ST] section 6.4.1 describes the Security Administrator role and identifies that the Security Administrator is able to perform all management functions. There are no restrictions involving administration of the TOE.
[ST] section 6.4.1 states, “The Security Administrator is associated with the defined login class “security-admin”, which has the necessary permission set to permit the administrator to perform all tasks necessary to manage Junos OS in accordance with the requirements of NDcPP. “

3.4.5.2 Guidance Documentation

172 The evaluator shall review the guidance documentation to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration.

Findings
PASS
[AGD] section “Understanding Roles and Services for Junos OS in Common Criteria and FIPS” describes that the Security Administrator has permissions to perform all tasks necessary to manage the Junos OS.
[AGD] sections “Configuring a Network Device collaborative Protection Profile for an Authorized Administrator” and “Configuring SSH on the Evaluated Configuration” describe supported public keys and algorithms that are required to be supported by the client for remote administration.

3.4.5.3 Tests

173 In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this cPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH, if the TSF shall be validated against the Functional Package for Secure Shell referenced in Section 2.2 of the cPP; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team’s test activities.

Note	No separate testing for FMT_SMR.2 is required.

3.5 Protection of the TSF (FPT)

3.5.1 FPT_SKP_EXT.1 Protection of TSF Data (for Reading of All Symmetric Keys)

3.5.1.1 TSS

174 The evaluator shall examine the TSS to determine that it details how any preshared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through any interface designed specifically for that purpose, by any enabled role, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

Findings
PASS
[ST] section 6.5.1 describes that all cryptographic keys are protected through kernel-level file access rights and the TOE does not permit viewing of cryptographic keys. Security Administrators do not have root access rights to the kernel.

3.5.1.2 Guidance Documentation

175 None

3.5.1.3 Tests

176 None

3.5.2 FPT_STM_EXT.1 Reliable Time Stamps

3.5.2.1 TSS

177 The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time, and that it provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

Findings
PASS
[ST] section 6.5.1 describes that the TOE supports synchronization with an NTP server or its own clock function within Junos OS that is maintained using the hardware Time Stamp Counter as the clock source and lists each security function that makes use of time.

178 If “obtain time from the underlying virtualization system” is selected, the evaluator shall examine the TSS to ensure that it identifies the VS interface the TOE uses to obtain time. If there is a delay between updates to the time on the VS and updating the time on the TOE, the TSS shall identify the maximum possible delay.

EA Not Applicable	The ST does not select “obtain time from the underlying virtualization system.”

3.5.2.2 Guidance Documentation

179 The evaluator shall examine the guidance documentation to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the guidance documentation instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

Findings
PASS
[ST] section “Customize Time” instructs the administrator how to set the time and the configuration of the NTP client.

180 If the TOE supports obtaining time from the underlying VS, the evaluator shall verify the Guidance Documentation specifies any configuration steps necessary. If no configuration is necessary, no statement is necessary in the Guidance Documentation. If there is a delay between updates to the time on the VS and updating the time on the TOE, the evaluator shall ensure the Guidance Documentation informs the administrator of the maximum possible delay.

EA Not Applicable	The ST does not select “obtain time from the underlying virtualization system.”

3.5.2.3 Tests

181 The evaluator shall perform the following tests:

a. Test 1: If the TOE supports direct setting of the time by the Security Administrator then the evaluator shall use the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

High-Level Test Description
Change the date and time to times in the past using the CLI.
Findings
PASS – The evaluator confirmed the time was set correctly.

b. Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE and set up a communication path with the NTP server. The evaluator shall observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the guidance documentation.

Note	This test is performed in FCS_NTP_EXT.1.1 Test 1 and FCS_NTP_EXT.1.2 Test 1/Known-Good NTP usage.

c. Test 3 [conditional]: If the TOE obtains time from the underlying VS, the evaluator shall record the time on the TOE, modify the time on the underlying VS, and verify the modified time is reflected by the TOE. If there is a delay between the setting the time

on the VS and when the time is reflected on the TOE, the evaluator shall ensure this delay is consistent with the TSS and Guidance.

Test Not Applicable	The TOE does not obtain time from the underlying VS.

182 If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.

Test Not Applicable	The audit component of the TOE does not consist of several parts with independent time information.

3.5.3 FPT_TST_EXT.1 TSF Testing

3.5.3.1 TSS

NIAP TD0836

183 The evaluator shall examine the TSS to ensure that it details each of the self-tests that are identified by the SFR; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly. If more than one failure response is listed in FPT_TST_EXT.1.2, the evaluator shall examine the TSS to ensure it clarifies which response is associated with which type of failure.

Findings
PASS
[ST] section 6.5.2 describes each self-test run by the TSF and what they are doing. The TSS also provides an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

184 For distributed TOEs the evaluator shall examine the TSS to ensure that it details which TOE component performs which self-tests and when these self-tests are run. The evaluator shall also examine the TSS to ensure it describes how the TOE reacts if one or more TOE components fail self-testing (e.g. halting and displaying an error message; failover behaviour).

EA Not Applicable	The TOE is not distributed.

3.5.3.2 Guidance Documentation

185 The evaluator shall also ensure that the guidance documentation describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

Findings
PASS

[AGD] section “Understanding FIPS Self-Tests” describes that if any of the self-tests fail the TOE will enter FIPS error state (panic) and reboot. These errors correspond to those described in the TSS.

186 For distributed TOEs the evaluator shall ensure that the guidance documentation describes how to determine from an error message returned which TOE component has failed the self-test.

EA Not Applicable	The TOE is not distributed.

3.5.3.3 Tests

NIAP TD0836

187 It is expected that at least the following tests are performed:

1 Verification of the integrity of the firmware and executable software of the TOE

2 Verification of the correct operation of the cryptographic functions necessary to fulfil any of the SFRs.

188 Although formal compliance is not mandated, the self-tests performed should aim for a level of confidence comparable to:

a. [FIPS 140-2], chap. 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. Note that the testing is not restricted to the cryptographic functions of the TOE.

b. [FIPS 140-2], chap. 4.9.1, Cryptographic algorithm test for the verification of the correct operation of cryptographic functions. Alternatively, national requirements of any CCRA member state for the security evaluation of cryptographic functions should be considered as appropriate.

189 The evaluator shall verify that the self-tests described above are carried out according to the SFR and in agreement with the descriptions in the TSS.

High-Level Test Description
Restart the TOE and verify that the startup includes an indicator that self-tests were executed and passed permitting the device to operate.
Findings
PASS – The evaluator confirmed TOE displays the boot process including self-tests.

190 For distributed TOEs the evaluator shall perform testing of self-tests on all TOE components according to the description in the TSS about which self-test are performed by which component.

Test Not Applicable	The TOE is not distributed.

3.5.4 FPT_TUD_EXT.1 Trusted Update

3.5.4.1 TSS

191 The evaluator shall verify that the TSS describe how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the TSS shall describe how and when the inactive version becomes active. The evaluator shall verify this description.

Findings
PASS
[ST] section 6.5.3 states that the Security Administrator can query the current version of the TOE software using the CLI command “show version local”.

192 The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system firmware and software (for simplicity the term 'software' will be used in the following although the requirements apply to firmware and software). The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. The evaluator shall verify that the TSS describes the method by which the digital signature is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature of the update, and the actions that take place for both successful and unsuccessful signature verification.

Findings
PASS
[ST] section 6.5.3 describes all TSF software update mechanisms for updating the system. The TOE updates are downloaded and applied manually. The TOE verifies the update’s manifest is digitally signed and will only process the update if the signature is confirmed. If the signature is verified the manifest is stored and each file is verified against the manifest before being executed. If any of the fingerprints in an update are not correctly verified, the TOE reverts to the last known good image.

193 If the options ‘support automatic checking for updates’ or ‘support automatic updates’ are chosen from the selection in FPT_TUD_EXT.1.2, the evaluator shall verify that the TSS explains what actions are involved in automatic checking or automatic updating by the TOE, respectively.

EA Not Applicable	The TOE does not support automatic checking or automatic updates.

194 For distributed TOEs, the evaluator shall examine the TSS to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component. Alternatively, this description can be provided in the guidance documentation. In that case the evaluator shall examine the guidance documentation instead.

EA Not Applicable	The TOE is not distributed.

3.5.4.2 Guidance Documentation

195 The evaluator shall verify that the guidance documentation describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the guidance documentation shall describe how to query the loaded but inactive version.

Findings

PASS

[AGD] section “Enabling FIPS Mode” describes that the TOE version can be queried with ‘show version local’ when the TOE is in FIPS mode.

196 The evaluator shall verify that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification). The description shall include the procedures for successful and unsuccessful verification. The description shall correspond to the description in the TSS.

Findings

PASS

[AGD] section “Install Junos OS Software Package” describes that the verification of authenticity of the update is performed using digital signatures.

If the verification is successful, the administrator is instructed to reboot the device to load the installation.

[AGD] section “Install Junos OS Software Package” states, “The installation proceeds only if the verification is successful.”

If the verification is unsuccessful the TOE version is not updated.

197 For distributed TOEs the evaluator shall verify that the guidance documentation describes how the versions of individual TOE components are determined for FPT_TUD_EXT.1, how all TOE components are updated, and the error conditions that may arise from checking or applying the update (e.g. failure of signature verification, or exceeding available storage space) along with appropriate recovery actions. The guidance documentation only has to describe the procedures relevant for the Security Administrator; it does not need to give information about the internal communication that takes place when applying updates.

EA Not Applicable The TOE is not distributed.

198 If this was information was not provided in the TSS: For distributed TOEs, the evaluator shall examine the Guidance Documentation to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component.

EA Not Applicable The TOE is not distributed.

199 If this was information was not provided in the TSS: If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the Guidance Documentation contains a description of how the

certificates are contained on the device. The evaluator shall also ensure that the Guidance Documentation describes how the certificates are installed/updated/selected, if necessary.

EA Not Applicable	The ST does not claim a certificate-based mechanism for software updates.

3.5.4.3 Tests

200 The evaluator shall perform the following tests:

3 Test 1: The evaluator shall perform the version verification activity to determine the current version of the product. If a trusted update can be installed on the TOE with a delayed activation, the evaluator shall also query the most recently installed version (for this test the TOE shall be in a state where these two versions match). The evaluator obtains a legitimate update using procedures described in the guidance documentation and verifies that it is successfully installed on the TOE. For some TOEs loading the update onto the TOE and activation of the update are separate steps ('activation' could be performed e.g. by a distinct activation step or by rebooting the device). In that case, the evaluator shall verify after loading the update onto the TOE but before activation of the update that the current version of the product did not change but the most recently installed version has changed to the new product version. After the update, the evaluator shall perform the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again.

High-Level Test Description
Get the current version of the TOE. Attempt to install a valid update using the CLI. Verify the installation succeeds and the version is updated.
Findings
PASS - The evaluator confirmed the TOE displayed its current version, successfully installed a valid update, and displayed the updated version.

4 Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator shall first confirm that no updates are pending and then perform the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator shall obtain or produce illegitimate updates as defined below and attempt to install them on the TOE. The evaluator shall verify that the TOE rejects all of the illegitimate updates. The evaluator shall perform this test using all of the following forms of illegitimate updates:

i. A modified version (e.g. using a hex editor) of a legitimately signed update

ii. An image that has not been signed

iii. An image signed with an invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate signature)

iv. The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify that both the current version and most recently installed version, reflect the same version information as prior to the update attempt.

High-Level Test Description
Get the current version of the TOE. Attempt to install updates with a bad signature (modified update), without a signature, and with an untrusted signature (untrusted key) using the CLI interface. Verify the installation fails and the version has not changed.
Findings
PASS – The evaluator confirmed that the TOE rejects all of the illegitimate updates.

201 The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automatic checking for updates, automatic updates).

202 For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.

3.6 TOE Access (FTA)

3.6.1 FTA_SSL.3 TSF-Initiated Termination

3.6.1.1 TSS

203 The evaluator shall examine the TSS to determine that it details the administrative remote session termination and the related inactivity time period.

Findings
PASS
[ST] section 6.6.1 describes that the Security Administrator can configure the inactivity time period for remote session termination.

3.6.1.2 Guidance Documentation

204 The evaluator shall confirm that the guidance documentation includes instructions for configuring the inactivity time period for remote administrative session termination.

Findings
PASS

[AGD] section “Configure Session Termination” provides instruction for configuring the inactivity time period for remote administrative session termination.

3.6.1.3 Tests

205 For each method of remote administration, the evaluator shall perform the following test:

a. Test 1: The evaluator shall follow the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a remote interactive session with the TOE. The evaluator shall then observe that the session is terminated after the configured time period.

High-Level Test Description
Configure the inactivity timeout to several values. Login to each remote interface and verify the TOE terminates the session when the inactivity timer has expired.
Findings
PASS – The evaluator confirmed that the session is terminated after the configured time period.

3.6.2 FTA_SSL.4 User-Initiated Termination

3.6.2.1 TSS

206 The evaluator shall examine the TSS to determine that it details how the remote administrative session (and if applicable the local administrative session) are terminated.

Findings
PASS
[ST] section 6.6.1 describes that local and remote administrative sessions can be terminated by the administrative user by typing ‘exit’.

3.6.2.2 Guidance Documentation

207 The evaluator shall confirm that the guidance documentation states how to terminate a remote interactive session (and if applicable the local administrative session).

Findings
PASS
[AGD] section “Sample Output for User Initiated Termination” shows how to terminate a remote and local administrative session using ‘exit’.

3.6.2.3 Tests

208 The evaluator shall perform the following tests:

a. Test 1 [conditional]: If the TOE supports local administration, the evaluator shall initiate an interactive local session with the TOE. The evaluator shall then follow the guidance

documentation to exit or log off the session and observes that the session has been terminated.

High-Level Test Description
Log into the Local Console. Log out of the Local Console session.
Findings
PASS – The evaluator confirmed the local session was successfully terminated.

b. Test 2: For each method of remote administration, the evaluator shall initiate an interactive remote session with the TOE. The evaluator shall then follow the guidance documentation to exit or log off the session and observes that the session has been terminated.

High-Level Test Description
Log into the SSH and Web UI interfaces. Log out of each session.
Findings
PASS – The evaluator confirmed interactive session was successfully terminated.

3.6.3 FTA_TAB.1 Default TOE Access Banners

3.6.3.1 TSS

209 The evaluator shall check the TSS to ensure that it details each administrative method of access (local and/or remote) available to the Security Administrator (e.g. serial port, SSH, HTTPS). The evaluator shall check the TSS to ensure that all administrative methods of access available to the Security Administrator are listed and that the TSS states that the TOE is displaying an advisory notice and a consent warning message for each administrative method of access. The advisory notice and the consent warning message might be different for different administrative methods of access and might be configured during initial configuration (e.g. via configuration file).

Findings
PASS
[ST] section 6.6.1 describes that the access banner is displayed for all administrative access methods (SSH and local interface).

3.6.3.2 Guidance Documentation

210 The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message.

Findings
PASS

[AGD] section “Configuring a System Login Message and Announcement” describes how to configure the banner login message.

3.6.3.3 Tests

211 The evaluator shall also perform the following test:

a. Test 1: The evaluator shall follow the guidance documentation to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

High-Level Test Description
Configure a notice and consent warning message. Verify the notice and consent warning message is displayed prior to establishing an administrative session at the Web UI, Remote CLI, and Local Console.
Findings
PASS – The evaluator confirmed the notice and consent warning message is displayed in each instance.

3.7 Trusted path/channels (FTP)

3.7.1 FTP_ITC.1 Inter-TSF trusted channel

3.7.1.1 TSS

212 The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each secure communication mechanism is identified in terms of the allowed protocols for that IT entity, whether the TOE acts as a server or a client, and the method of assured identification of the non-TSF endpoint. The evaluator shall also confirm that all secure communication mechanisms are described in sufficient detail to allow the evaluator to match them to the cryptographic protocol Security Functional Requirements listed in the ST.

Findings
PASS
[ST] section 6.7.1 describes that the TOE acts as a SSH server for all communications with a remote audit server. The evaluator confirmed all secure communication mechanisms are described in sufficient detail to allow them to be matched to the cryptographic protocol.

3.7.1.2 Guidance Documentation

213 The evaluator shall confirm that the guidance documentation contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

Findings
PASS

[AGD] section “Syslog Server Configuration on a Linux System” provides instructions for establishing the allowed SSH protocols with the audit server.

[AGD] states. “NOTE: For SSH connection being unintentionally broken, we need to re-initiate the SSH connection to log in back to TOE.”

3.7.1.3 Tests

214 The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report.

215 The evaluator shall perform the following tests:

a. Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

Note	The TOE maintains a trusted channel to the remote audit server, which is set up as per the evaluated configuration. It is constantly tested throughout the evaluation. The trusted channel is specifically tested as part of FCS_SSH_EXT.1.

b. Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE.

Test Not Applicable	The TOE does not initiate the trusted channel.

c. Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.

Note	The channel data for the SSH trusted channel to the remote audit server was verified to be encrypted as part of FAU_STG_EXT.1 Test 1.

d. Test 4: Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route to the external IT entities.

The evaluator shall, for each instance where the TOE acts as a client utilizing a secure communication mechanism with a distinct IT entity, interrupt the connection of that IT entity for the following durations: i) a duration that exceeds the TOE’s application layer timeout setting, ii) a duration shorter than the application layer timeout but of sufficient length to interrupt the network link layer.

The evaluator shall ensure that, when the connectivity is restored, communications are appropriately protected and no TSF data is sent in plaintext.

In the case where the TOE is able to detect TOE external interruption (such as a cable being physically removed or a virtual connection being disabled), another network

device shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall be external to the TOE (i.e., by manipulating the test environment and not by TOE configuration change).

Test Not Applicable	The TOE does not act as a client utilizing a secure communication mechanism with a distinct IT entity.

216 Further assurance activities are associated with the specific protocols.

Note	The trusted channel is specifically tested by FCS_SSH_EXT.1.

217 For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of external secure channels to TOE components in the Security Target.

Test Not Applicable	The TOE is not distributed.

218 The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report.

3.7.2 FTP_TRP.1/Admin Trusted Path

3.7.2.1 TSS

219 The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

Findings
PASS
[ST] section 6.7.1 describes that the TOE acts as a SSH server for remote administration. The evaluator confirmed all protocols listed in the TSS are consistent with those in the requirement and included in the requirements in the ST.

3.7.2.2 Guidance Documentation

220 The evaluator shall confirm that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method.

Findings
PASS
[AGD] sections “Configuring Security Administrator and FIPS User Identification and Access”, “Configuring a Network Device collaborative Protection Profile for an Authorized Administrator” and “Configuring SSH on the Evaluated Configuration” describes all the preparatory steps necessary to

configure the TOE and remote administrative SSH protocol for establishing the remote administrative sessions.

3.7.2.3 Tests

221 The evaluator shall perform the following tests:

a. Test 1: The evaluators shall ensure that communications using each specified (in the guidance documentation) remote administration method is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

Note	The SSH Remote CLI trusted path is set up as per the evaluated configuration. It is constantly tested throughout the evaluation. The SSH protocol is tested in FCS_SSH_EXT.1.

b. Test 2: The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext.

Note	FCS_SSH_EXT.1 testing shows the TOE successfully establishing trusted paths. The remote trusted path client is a known good SSH client implementation, so the successful transfer of channel data shows the channel data is not sent in plaintext (i.e., the client would terminate the connection due to decryption and/or integrity errors if the data was sent in plaintext).

222 Further assurance activities are associated with the specific protocols.

Note	The SSH Remote CLI trusted path is set up as per the evaluated configuration. It is constantly tested throughout the evaluation. The SSH protocol is tested in FCS_SSH_EXT.1.

223 For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.

Test Not Applicable	The TOE is not distributed.

4 Evaluation Activities for Optional Requirements

4.1 Security Audit (FAU)

4.1.1 FAU_STG.1 Protected Audit Trail Storage

4.1.1.1 TSS

224 The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally and how these records are protected against unauthorized modification or deletion. The evaluator shall ensure that the TSS describes the conditions that must be met for authorized deletion of audit records.

Findings
PASS
[ST] section 6.1.1 describes that the TOE has an active log file of a maximum size of 1Gb and when the active file reaches its maximum size it is compressed and archived. Only a Security Administrator can read or delete logs and archive files through the CLI interface or through direct access to the filesystem.

225 For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how local storage is implemented among the different TOE components (e.g. every TOE component does its own local storage or the data is sent to another TOE component for central local storage of all audit events).

EA Not Applicable	The TOE is not distributed.

4.1.1.2 Guidance Documentation

226 The evaluator shall examine the guidance documentation to determine that it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion.

Findings
PASS
There is no configuration required for protection of locally stored audit data against unauthorized modification or deletion. [ST] section 6.1.1 states, only a Security Administrator can read or delete logs and archive files through the CLI interface or through direct access to the filesystem.

4.1.1.3 Tests

227 The evaluator shall perform the following tests:

a. Test 1: The evaluator shall attempt to access the audit trail without authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all) and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail. According to the

implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to access the audit trail can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

High-Level Test Description
As a user without Security Administrator permissions attempt to delete a log file. Confirm that the attempt fails.
Findings
PASS – The evaluator confirmed that a non-administrative user is unable to delete audit records.

b. Test 2: The evaluator shall access the audit trail as an authenticated Security Administrator and attempt to delete the audit records (if supported by the TOE, and to the extent described in the TSS). The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records authorized for deletion are deleted.

High-Level Test Description
As the Security Administrator, attempt to delete a log file. The attempt will be successful.
Findings
PASS – The evaluator confirmed the records were successfully deleted by the Security Administrator.

228 For distributed TOEs the evaluator shall perform test 1 and test 2 for each component that is defined by the TSS to be covered by this SFR.

Test Not Applicable	The TOE is not distributed.

5 Evaluation Activities for Selection-Based Requirements

5.1 Cryptographic Support (FCS)

5.1.1 FCS_NTP_EXT.1 NTP Protocol

5.1.1.1 TSS

FCS_NTP_EXT.1.1

229 The evaluator shall examine the TSS to ensure it identifies the version of NTP supported, how it is implemented and what approach the TOE uses to ensure the timestamp it receives from an NTP timeserver (or NTP peer) is from an authenticated source and the integrity of the time has been maintained.

The TOE must support at least one of the methods or may use multiple methods, as specified in the SFR element 1.2. The evaluator shall ensure that each method selected in the ST is described in the TSS, including the version of NTP supported in element 1.1, the message digest algorithms used to verify the authenticity of the timestamp and/or the protocols used to ensure integrity of the timestamp.

Findings
PASS
[ST] section 6.2.2 describes that the TOE supports NTPv4 and uses symmetric keys with SHA-256 message digest for authentication and integrity.

FCS_NTP_EXT.1.2, FCS_NTP_EXT.1.3, and FCS_NTP_EXT.1.4

230 None.

5.1.1.2 Guidance Documentation

FCS_NTP_EXT.1.1

231 The evaluator shall examine the guidance documentation to ensure it provides the Security Administrator instructions as how to configure the version of NTP supported, how to configure multiple NTP servers for the TOE's time source and how to configure the TOE to use the method(s) that are selected in the ST.

Findings
PASS
[AGD] section "Customize Time" provides instructions how to configure multiple NTP servers for the TOE's time source and how to configure the method selected in the ST. Note that the TOE only supports NTPv4 in fips mode.

FCS_NTP_EXT.1.2

232 For each of the secondary selections made in the ST, the evaluator shall examine the guidance document to ensure it instructs the Security Administrator how to configure the TOE to use the algorithms that support the authenticity of the timestamp and/or how to configure the TOE to use the protocols that ensure the integrity of the timestamp.

Assurance Activity Note:

Each primary selection in the SFR contains selections that specify a cryptographic algorithm or cryptographic protocol. For each of these secondary selections made in the ST, the evaluator shall examine the guidance documentation to ensure that the documentation instructs the Security Administrator how to configure the TOE to use the chosen option(s).

Findings
PASS
[AGD] section “Customize Time” provides instructions how to configure the TOE to use the algorithms that support the authenticity of the timestamp.

FCS_NTP_EXT.1.3

233 The evaluator shall examine the guidance documentation to ensure it provides the Security Administrator instructions as how to configure the TOE to not accept broadcast and multicast NTP packets that would result in the timestamp being updated.

Findings
PASS
The guidance does not require configuration for TOE to not accept broadcast and multicast NTP packets.

FCS_NTP_EXT.1.4

234 None.

5.1.1.3 Tests

FCS_NTP_EXT.1.1

235 The version of NTP selected in element 1.1 and specified in the ST shall be verified by observing establishment of a connection to an external NTP server known to be using the specified version(s) of NTP. This may be combined with tests of other aspects of FCS_NTP_EXT.1 as described below.

High-Level Test Description
Configure the TOE to sync its time with an NTP server. Check the version matches the one claimed.
Findings
PASS – The evaluator confirmed the TOE supports NTPv4.

FCS_NTP_EXT.1.2

236 The cryptographic algorithms selected in element 1.2 and specified in the ST will have been specified in an FCS_COP SFR and tested in the accompanying Evaluation Activity for that SFR. Likewise, the cryptographic protocol selected in in element 1.2 and specified in the ST will have been specified in an FCS SFR and tested in the accompanying Evaluation Activity for that SFR.

[conditional] If the message digest algorithm is claimed in element 1.2, the evaluator shall change the message digest algorithm used by the NTP server in such a way that the new value does not match the configuration on the TOE and confirms that the TOE does not synchronize to this time source.

High-Level Test Description
Using a custom tool, force the NTP server to respond with a MAC which uses the wrong hashing algorithm.
Findings
PASS – The evaluator confirmed the TOE denies the NTP server's response hashed with an incorrect digest algorithm.

The evaluator shall use a packet sniffer to capture the network traffic between the TOE and the NTP server. The evaluator shall use the captured network traffic, to verify the NTP version, to observe time change of the TOE and uses the TOE's audit log to determine that the TOE accepted the NTP server's timestamp update.

The captured traffic is also used to verify that the appropriate message digest algorithm was used to authenticate the time source and/or the appropriate protocol was used to ensure integrity of the timestamp that was transmitted in the NTP packets.

High-Level Test Description
Configure the NTP server to have a different time than the TOE. Ensure the TOE uses NTP to synchronize to the NTP server. Verify the correct version and hashing algorithm is being used. Show that the TOE is accepting the time from the NTP server.
Findings
PASS – The evaluator confirmed the TOE accepts the NTP server's response signed with a valid digest algorithm.

FCS_NTP_EXT.1.3

237 The evaluator shall configure NTP server(s) to support periodic time updates to broadcast and multicast addresses. The evaluator shall confirm the TOE is configured to not accept broadcast and multicast NTP packets that would result in the timestamp being updated. The evaluator shall check that the time stamp is not updated after receipt of the broadcast and multicast packets.

High-Level Test Description
Configure the NTP server in the environment to only transmit broadcast and multicast. Toggle the NTP client functionality on the TOE and show that the TOE does not synchronize to the NTP server in the environment.

Findings
PASS – The evaluator confirmed the TOE does not accept broadcast and multicast NTP packets and the time stamp was not updated.

FCS_NTP_EXT.1.4

238 The evaluator shall perform the following tests:

a. Test 1: The evaluator shall confirm the TOE supports configuration of at least three (3) NTP time sources. The evaluator shall configure at least three NTP servers to support periodic time updates to the TOE. The evaluator shall confirm the TOE is configured to accept NTP packets that would result in the timestamp being updated from each of the NTP servers. The evaluator shall check that the time stamp is updated after receipt of the NTP packets. The purpose of this test to verify that the TOE can be configured to synchronize with multiple NTP servers. It is up to the evaluator to determine that the multi-source update of the time information is appropriate and consistent with the behaviour prescribed by the RFC 1305 for NTPv3 and RFC 5905 for NTPv4.

High-Level Test Description
Configure three NTP time servers. When configuring them, ensure that only one IP is active at any given time and show that the TOE is capable of synchronizing to it. Remove all three NTP time servers when complete to show conformance with auditing requirements.
Findings
PASS – The evaluator confirmed the TOE supports configuration of at least three (3) NTP time sources.

b. Test 2: (The intent of this test is to ensure that the TOE would only accept NTP updates from configured NTP Servers). The evaluator shall confirm that the TOE would not synchronize to other, not explicitly configured time sources by sending an otherwise valid but unsolicited NTP Server response indicating different time from the TOE's current system time. This rogue time source needs to be configured in a way (e.g. degrade or disable valid and configured NTP servers) that could plausibly result in unsolicited updates becoming a preferred time source if they are not discarded by the TOE. The TOE is not mandated to respond in a detectable way or audit the occurrence of such unsolicited updates. It is up to the evaluator to craft and transmit unsolicited updates in a way that would be consistent with the behaviour of a correctly-functioning NTP server.

High-Level Test Description
Using a custom tool, transmit legitimate NTP server responses such that the NTP client could theoretically respond. Show that the TOE ignores these responses because they are spoofed.
Findings
PASS – The evaluator confirmed the TOE does not synchronize to other, not explicitly configured time sources.

5.2 Identification and Authentication (FIA)

5.2.1 FIA_AFL.1 Authentication Failure Management

5.2.1.1 TSS

239 The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.

Findings
PASS
[ST] section 6.3.1 describes that the successive unsuccessful authentication attempts through SSH using username and password are detected following the first failed login attempt and the tries-before-disconnect are configurable between (2-10) attempts. The remote administrator is prevented from successfully logging in for an administrator configurable lockout-period (1-43,200 minutes).

240 The evaluator shall examine the TSS to confirm that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking).

Findings
PASS
[ST] section 6.3.1 describes that the administrator is always able to login through the serial console even if the account is locked out of remote access.

5.2.1.2 Guidance Documentation

241 The evaluator shall examine the guidance documentation to ensure that instructions for configuring the number of successive unsuccessful authentication attempts and time period (if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each “action” specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.

Findings
PASS
[AGD] section “Limiting the Number of User Login Attempts for SSH Sessions” provides instructions for configuring the number of successive unsuccessful authentication attempts and lockout time period.

242 The evaluator shall examine the guidance documentation to confirm that it describes, and identifies the importance of, any actions that are required in order to ensure that administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA_AFL.1.

Findings
PASS
[AGD] section “Limiting the Number of User Login Attempts for SSH Session” describes that the local administrator access will be maintained even if the remote administration is made permanently or temporarily unavailable due to the multiple failed login attempts.

5.2.1.3 Tests

243 The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g. any passwords entered as part of establishing the connection protocol or the remote administrator application):

a. Test 1: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE (and, if the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall also use the operational guidance to configure the time period after which access is reenabled). The evaluator shall test that once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful.

Note	This test is performed in conjunction with FIA_AFL.1 Test 2.

b. Test 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows.

If the administrator action selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall confirm by testing that following the operational guidance and performing each action specified in the ST to re-enable the remote administrator’s access results in successful access (when using valid credentials for that administrator).

Test Not Applicable	Administrator action selection is not included in the ST.

If the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall wait for just less than the time period configured in Test 1 and show that an authorisation attempt using valid credentials does not result in successful access. The evaluator shall then wait until just after the time period configured in Test 1 and show that an authorisation attempt using valid credentials results in successful access.

High-Level Test Description
For the remote CLI, attempt to login using an incorrect password until the login failure threshold has been met. Shortly before the lockout duration has expired, attempt to login using the correct password and verify the login attempt fails. After the lockout duration has expired, attempt to login using the correct password and verify the login attempt succeeds.
Findings
PASS – The evaluator confirmed once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful. The evaluator confirmed the TOE successfully enforces time period configuration for lockout.

5.2.2 FIA_UAU.7 Protected Authentication Feedback

5.2.2.1 TSS

244 None

5.2.2.2 Guidance Documentation

245 The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps to ensure authentication data is not revealed while entering for each local login allowed.

Findings
PASS
No configuration is necessary to ensure authentication data is not revealed while entering for each local login allowed.

5.2.2.3 Tests

246 The evaluator shall perform the following test for each method of local login allowed:

a. Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.

Note	This test was conducted as part of FIA_UIA_EXT.1 Test 1 Step 6 which showed at most obscured feedback was provided while entering authentication information at the local console.

5.2.3 FIA_PMG_EXT.1 Password Management

5.2.3.1 TSS

247 The evaluator shall check that the TSS:

a. lists the supported special character(s) for the composition of administrator passwords

b. to ensure that the minimum_password_length parameter is configurable by a Security Administrator.

c. lists the range of values supported for the minimum_password_length parameter. The listed range shall include the value of 15.

Findings
PASS
[ST] section 6.3.1 contains the list of all the supported special characters and states that minimum password lengths are configurable between 10-20 characters. The evaluator confirmed the list of special characters matches FIA_PMG_EXT.1.1.

5.2.3.2 Guidance Documentation

248 The evaluator shall examine the guidance documentation to determine that it:

a. identifies the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords, and

b. provides instructions on setting the minimum password length and describes the valid minimum password lengths supported.

Findings
PASS
[AGD] section “Understanding the Associated Password Rules for an Authorized Administrator” identifies all the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords. The section also describes how to set the minimum password length and the password lengths supported between 10 and 20 characters.

5.2.3.3 Tests

249 The evaluator shall perform the following tests.

a. Test 1: The evaluator shall compose passwords that meet the requirements in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, and a minimum length listed in the requirement are supported and justify the subset of those characters chosen for testing.

High-Level Test Description
Set the minimum password length to 10 characters. Set the password for an account and verify authentication succeeds using the configured password using passwords that are 10 characters long and passwords that contain all of the claimed characters. Verify the passwords can be successfully set and used to login.
Findings
PASS – The evaluator confirmed TOE successfully enforces password requirements.

b. Test 2: The evaluator shall compose passwords that do not meet the requirements in some way. For each password, the evaluator shall verify that the TOE does not support the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that the TOE enforces the allowed characters and the minimum length listed in the requirement and justify the subset of those characters chosen for testing.

High-Level Test Description
Using each management interface, attempt to set a password whose length is one less than the configured password policy. Verify the password change is rejected.
Findings

PASS – The evaluator confirmed TOE successfully denies unsupported password requirements.

5.3 Protection of the TSF (FPT)

5.3.1 FPT_APW_EXT.1 Protection of Administrator Passwords

5.3.1.1 TSS

250 The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.

Findings

PASS

[ST] section 6.5.1 describes that administrator passwords are stored hashed using sha512 and that there are no interfaces available that are designed specifically for the purpose of viewing passwords in plaintext.

5.3.1.2 Guidance Documentation

251 None

5.3.1.3 Tests

252 None

5.4 Security management (FMT)

5.4.1 FMT_MOF.1/Functions Management of Security Functions Behaviour

5.4.1.1 TSS

253 For distributed TOEs see [ND-SD] Section 2.4.1.1.

EA Not Applicable The TOE is not distributed.

254 For non-distributed TOEs, the evaluator shall ensure the TSS for each administrative function identified the TSS details how the Security Administrator determines or modifies the behaviour of (whichever is supported by the TOE) transmitting audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full (whichever is supported by the TOE).

Findings

PASS

[ST] section 6.4.1 describes that the TOE modifies the behaviour of transmission of audit data to an external IT entity by modifying the configuration of SSH trusted channel.

The TOE modifies the behaviour of handling of audit data by setting log file size limits.

5.4.1.2 Guidance Documentation

255 For distributed TOEs see [ND-SD] Section 2.4.1.2.

EA Not Applicable The TOE is not distributed.

256 For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the Security Administrator determines or modifies the behaviour of (whichever is supported by the TOE) transmitting audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full (whichever is supported by the TOE) are performed to include required configuration settings.

Findings

PASS

[AGD] section “Configuring Event Logging to a Remote Server when Initiating the Connection from the Remote Server” describes how the Security Administrator modifies the behaviour of transmitting audit data to an external IT entity.

[AGD] section “Configuring Audit Log Options in the Evaluated Configuration” describes how the Security Administrator modifies the handling of audit data.

5.4.1.3 Tests

257 **If ‘transmission of audit data to external IT entity’ is selected from the second selection together with 'modify the behaviour of' in the first selection**

258 The evaluator shall perform the following tests:

a. Test 1: The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to modify the security related parameters can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

High-Level Test Description

As a user without Security Administrator permissions attempt to modify all security related parameters for the transmission protocol for transmission of audit data to an external IT entity. Confirm that the attempts to modify the parameters fail.

Findings

PASS – The evaluator confirmed the non-administrative user is unable to modify parameters for the configuration of the transmission protocol.

b. Test 2: The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity with prior authentication as Security Administrator. The effects of the modifications should be confirmed.

The evaluator does not have to test all possible values of the security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity but at least one allowed value per parameter.

High-Level Test Description
As the Security Administrator, attempt to modify all security related parameters for the transmission protocol for transmission of audit data to an external IT entity. Confirm that the attempts to modify the parameters are successful.
Findings
PASS - The evaluator confirmed the Security Administrator is authorized to modify the security related parameters for the configuration of the transmission protocol.

259 **If 'handling of audit data' is selected from the second selection together with 'modify the behaviour of' in the first selection**

260 The evaluator shall perform the following tests:

a. Test 1: The evaluator shall try to modify all security related parameters for configuration of the handling of audit data without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator. The term ‘handling of audit data’ refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.4, FAU_STG_EXT.1.5 and FAU_STG_EXT.2.

High-Level Test Description
As a user without Security Administrator permissions attempt to modify all security related parameters for the handling of audit data. Confirm that the attempts to modify the parameters fail.
Findings
PASS - The evaluator confirmed the non-administrative user is unable to modify security related parameters for configuration of the handling of audit data.

b. Test 2: The evaluator shall try to modify all security related parameters for configuration of the handling of audit data with prior authentication as Security Administrator. The effects of the modifications should be confirmed. The term ‘handling of audit data’ refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.4, FAU_STG_EXT.1.5 and FAU_STG_EXT.2.

The evaluator does not necessarily have to test all possible values of the security related parameters for configuration of the handling of audit data but at least one allowed value per parameter.

Note	FAU_STG_EXT.1 Test 4 shows the Security Administrator is able to modify the handling of audit data.

261 **If 'audit functionality when Local Audit Storage Space is full' is selected from the second selection together with 'modify the behaviour of' in the first selection**

262 The evaluator shall perform the following tests:

a. Test 1: The evaluator shall try to modify the behaviour when Local Audit Storage Space is full without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). This attempt should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

Test Not Applicable	The ST does not claim this functionality.

b. Test 2: The evaluator shall try to modify the behaviour when Local Audit Storage Space is full with prior authentication as Security Administrator. This attempt should be successful. The effect of the change shall be verified.

The evaluator does not necessarily have to test all possible values for the behaviour when Local Audit Storage Space is full but at least one change between allowed values for the behaviour.

Test Not Applicable	The ST does not claim this functionality.

263 **If in the first selection 'determine the behaviour of' has been chosen together with for any of the options in the second selection**

264 The evaluator shall perform the following tests:

a. Test 1: The evaluator shall try to determine the behaviour of all options chosen from the second selection without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). This can be done in one test or in separate tests. The attempt(s) to determine the behaviour of the selected functions without administrator authentication shall fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

Test Not Applicable	The ST does not claim this functionality.

b. Test 2: The evaluator shall try to determine the behaviour of all options chosen from the second selection with prior authentication as Security Administrator. This can be done in one test or in separate tests. The attempt(s) to determine the behaviour of the selected functions with Security Administrator authentication shall be successful.

Test Not **Applicable**	The ST does not claim this functionality.

5.4.2 FMT_MOF.1/Services Management of Security Functions Behaviour

5.4.2.1 TSS

265 For distributed TOEs see [ND-SD] Section 2.4.1.1.

EA Not Applicable	The TOE is not distributed.

266 For non-distributed TOEs, the evaluator shall ensure the TSS lists the services the Security Administrator is able to start and stop and how that operation is performed.

Findings
PASS
[ST] section 6.4.1 describes that the Security Administrator is able to start and stop the SSH service used for the trusted communication channel to the external syslog server and the remote administrative sessions via the CLI interface.

5.4.2.2 Guidance Documentation

267 For distributed TOEs see [ND-SD] Section 2.4.1.2.

EA Not Applicable	The TOE is not distributed.

268 For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the TSS lists the services the Security Administrator is able to start and stop and how that operation is performed.

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” describes enabling the Netconf service with ‘activate system services netconf ssh’ or disabling SSH service with ‘deactivate system services netconf ssh’.

5.4.2.3 Tests

269 The evaluator shall perform the following tests:

a. Test 1: The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) without prior authentication as Security Administrator (either by authenticating as a user with no administrator privileges, if possible, or without prior authentication at

all). The attempt to enable/disable this service/these services should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to enable/disable this service/these services can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

High-Level Test Description
As a user without Security Administrator permissions attempt to disable a TOE service. Confirm that the attempt fails.
Findings
PASS - The evaluator confirmed the TOE does not allow the user with no administrator privileges to enable or disable the TOE services.

b. Test 2: The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) with prior authentication as Security Administrator. The attempt to enable/disable this service/these services should be successful.

High-Level Test Description
As the privileged user, attempt to stop one of the predefined services. The attempt will be successful. Privileged user service actions are performed as part of other tests.
Findings
PASS - The evaluator confirmed the Security Administrator is authorized to enable/disable a service.

5.4.3 FMT_MTD.1/CryptoKeys Management of TSF Data

5.4.3.1 TSS

270 For distributed TOEs see [ND-SD] Section 2.4.1.1.

EA Not Applicable	The TOE is not distributed.

271 For non-distributed TOEs, the evaluator shall ensure the TSS lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and names the operations that are performed.

Findings
PASS
[ST] section 6.4.1 supports generating and deleting SSH keys.

5.4.3.2 Guidance Documentation

272 For distributed TOEs see [ND-SD] Section 2.4.1.2.

EA Not Applicable	The TOE is not distributed.

273 For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the operations are performed on the keys the Security Administrator is able to manage.

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” describes how the TOE generates SSH host keys.
[AGD] section “Username and Public Key Authentication over SSH” provides instructions on how the Security Administrator manages Public Keys for user authentication over SSH.
[AGD] section “Configure MACsec” includes instructions for managing PSKs/CAKs.

5.4.3.3 Tests

274 The evaluator shall perform the following tests:

a. Test 1: The evaluator shall try to perform at least one of the related actions (modify, delete, generate/import) without prior authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all). Attempts to perform related actions without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to manage cryptographic keys can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

High-Level Test Description
As a user without Security Administrator permissions attempt to generate a cryptographic SSH Hostkey. Confirm that the attempt fails.
Findings
PASS – The evaluator confirmed that a non-administrative user is unable to manage cryptographic keys.

b. Test 2: The evaluator shall try to perform at least one of the related actions with prior authentication as Security Administrator. This attempt should be successful.

High-Level Test Description
Attempt to generate a SSH Hostkey as the Security Administrator. Verify the generation succeeds.
Findings
PASS – The evaluator confirmed the cryptographic key can be managed by the Security Administrator.

5.5 TOE Access (FTA)

5.5.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

5.5.1.1 TSS

275 The evaluator shall examine the TSS to determine that it details whether local administrative session locking or termination is supported and the related inactivity time period settings.

Findings
PASS
[ST] section 6.6.1 describes that the TOE supports local administrative session termination after a period of inactivity.

5.5.1.2 Guidance Documentation

276 The evaluator shall confirm that the guidance documentation states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period.

Findings
PASS
[AGD] section “Configuring Inactivity Timeout Period Configuration, and Local and Remote Idle Session Termination” provides instruction for configuring the inactivity time period for local administrative session termination.

5.5.1.3 Tests

277 The evaluator shall perform the following test:

a. Test 1: The evaluator shall follow the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a local interactive session with the TOE. The evaluator shall then observe that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator shall then ensure that reauthentication is needed when trying to unlock the session.

High-Level Test Description
Configure the global and local console idle timeout settings to several different values while establishing Local Console sessions. Verify the Local Console session is terminated when the threshold is reached.
Findings
PASS – The evaluator confirmed the Local Console session is terminated when the threshold is reached.

6 Evaluation Activities for Security Assurance Requirements

6.1 ASE: Security Target Evaluation

6.1.1 General Evaluation Activities for TOE Summary Specification (ASE_TSS.1) for All TOEs

278 When evaluating a Security Target, the evaluator performs the work units as presented in the CEM. In addition, the evaluator shall ensure the content of the TSS in the ST satisfies the EAs specified in [ND-SD] Section 2 (Evaluation Activities for SFRs).

Findings
PASS
The ASE CEM work units are documented in the proprietary ETR. The TSS Evaluation Activities are documented throughout this report.

6.1.2 Additional Evaluation Activities for TOE Summary Specification (ASE_TSS.1) for Distributed TOEs

279 For distributed TOEs only the SFRs classified as ‘all’ have to be fulfilled by all TOE parts. The SFRs classified as ‘One’ or ‘Feature Dependent’ only have to be fulfilled by either one or some TOE parts, respectively. To make sure that the distributed TOE as a whole fulfills all the SFRs the following actions for ASE_TSS.1 have to be performed as part of ASE_TSS.1.1E.

ASE_TSS.1 element	Evaluator Action
ASE_TSS.1.1C	The evaluator shall examine the TSS to determine that it is clear which TOE components contribute to each SFR or how the components combine to meet each SFR. The evaluator shall verify the sufficiency to fulfil the related SFRs. This includes checking that the TOE as a whole fully covers all SFRs and that all functionality that is required to be audited is in fact audited regardless of the component that carries it out.

Findings
PASS – N/A
The TOE is not a distributed TOE.

6.2 ADV: Development

6.2.1 Basic Functional Specification (ADV_FSP.1)

280 The EAs for this assurance component focus on understanding the interfaces (e.g., application programming interfaces, command line interfaces, graphical user interfaces, network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in [ND-SD] Section 2, and in EAs for AGD, ATE and AVA SARs in other parts of [ND-SD] Section 5.

281 The EAs presented in this section address the CEM work units ADV_FSP.1-1, ADV_FSP.1-2, ADV_FSP.1-3, and ADV_FSP.1-5.

282 The EAs are reworded for clarity and interpret the CEM work units such that they will result in more objective and repeatable actions by the evaluator. The EAs in this SD are intended to ensure the evaluators are consistently performing equivalent actions.

283 The documents to be examined for this assurance component in an evaluation are therefore the Security Target, AGD documentation, and any required supplementary information required by the cPP: no additional “functional specification” documentation is necessary to satisfy the EAs. The interfaces that need to be evaluated are also identified by reference to the EAs listed for each SFR and are expected to be identified in the context of the Security Target, AGD documentation, and any required supplementary information defined in the cPP rather than as a separate list specifically for the purposes of CC evaluation. The direct identification of documentation requirements and their assessment as part of the EAs for each SFR also means that the tracing required in ADV_FSP.1.2D (work units ADV_FSP.1-4, ADV_FSP.1-6 and ADV_FSP.1-7) is treated as implicit and no separate mapping information is required for this element.

6.2.1.1 Evaluation Activity

284 *The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.*

285 In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g. audit review or performing updates). Explicitly labeling TSFI as security relevant or non-security relevant is not necessary. A TSFI is implicitly security relevant if it is used to satisfy an evaluation activity, or if it is identified in the ST or guidance documentation as adhering to the security policies (as presented in the SFRs). The intent is that these interfaces will be adequately tested and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied. According to the description above 'security relevant' corresponds to the combination of 'SFR-enforcing' and 'SFR-supporting' as defined in CC Part 3, paragraph 224 and 225.

286 The set of TSFI that are provided as evaluation evidence are contained in the Security Target and the guidance documentation.

Findings
PASS

From section 7.2.1 of the [NDcPP]: “For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation.”

The [ST] and the guidance documentation comprise the functional specification. The evaluator was able to perform the Evaluation Activities specified in the [ND-SD], so the evaluator concluded that the functional specification sufficiently describes the parameters, purpose, and method of use for each TSFI that is identified as being security relevant.

6.2.1.2 Evaluation Activity

287 *The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.*

Findings

PASS

Please see the previous work unit.

6.2.1.3 Evaluation Activity

288 *The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.*

289 The evaluator shall use the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in [ND-SD] Section 2, including the EAs associated with testing of the interfaces.

290 It should be noted that there may be some SFRs that do not have a TSFI that is explicitly “mapped” to invoke the desired functionality. For example, generating a random bit string or destroying a cryptographic key that is no longer needed are capabilities that may be specified in SFRs, but are not invoked by an interface.

291 The required EAs define the design and interface information required to meet ADV_FSP.1. If the evaluator is unable to perform some EA, then the evaluator shall conclude that an adequate functional specification has not been provided.

Findings

PASS

From section 7.2.1 of the [NDcPP]: “For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation.”

The [ST] and the guidance documentation comprise the functional specification. The interfaces are implicitly mapped to SFRs if they are used to satisfy an Evaluation Activity for a specific SFR. The evaluator was able to perform the Evaluation Activities specified in the [ND-SD]; the Findings for SFR related Evaluation Activities are the mapping of interfaces to SFRs.

6.3 AGD: Guidance Documents

292 It is not necessary for a TOE to provide separate documentation to meet the individual requirements of AGD_OPE and AGD_PRE. Although the EAs in this section are described

under the traditionally separate AGD families, the mapping between the documentation provided by the developer and AGD_OPE and AGD_PRE requirements may be many-to-many, as long as all requirements are met in documentation that is delivered to Security Administrators and users (as appropriate) as part of the TOE.

293 Note that additional Evaluation Activities for the guidance documentation in the case of a distributed TOE are defined in Appendix B.4.2.1. (in the NDcPP-SD)

6.3.1 Operational User Guidance (AGD_OPE.1)

294 The evaluator performs the CEM work units associated with the AGD_OPE.1 SAR. Specific requirements and EAs on the guidance documentation are identified (where relevant) in the individual EAs for each SFR. For the related evaluation activities, the evaluation evidence documents Security Target, AGD documentation (user guidance) and functional specification documentation (if provided) shall be used as input documents. Each input document is subject to ALC_CMS.1-2 requirements.

295 In addition, the evaluator performs the EAs specified below.

6.3.1.1 Evaluation Activity

296 *The evaluator shall ensure the Operational guidance documentation is distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.*

Findings
PASS
The guidance documentation is posted to the NIAP website, ensuring administrators are aware of the documentation. The guidance documentation is also posted to the vendor's website at https://www.juniper.net/.

6.3.1.2 Evaluation Activity

297 *The evaluator shall ensure that the Operational guidance is provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.*

Findings
PASS
There is only one operational environment claimed in [ST] section 2.4. All TOE platforms claimed in [ST] are covered by the operational guidance. This is evidenced by the platform equivalency.

6.3.1.3 Evaluation Activity

298 *The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.*

Findings
PASS
[AGD] sections “Enabling FIPS Mode”, “Configuring SSH and Console Connection” and “Configure MACsec” describe the configuration of the cryptographic operations (i.e., engines, protocols, algorithms, key sizes) to be consistent with the evaluated configuration. No configuration needed once the TOE is in the evaluated configuration. The [AGD] provides instructions for ensuring the evaluated functionality is used.

6.3.1.4 Evaluation Activity

299 *The evaluator shall ensure the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.*

Findings
PASS
The evaluator confirmed [AGD] covers configuration of the in-scope functionality where additional configuration might be required.

6.3.1.5 Evaluation Activity

300 In addition, the evaluator shall ensure that the following requirements are also met

5 The guidance documentation shall contain instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.

6 The evaluator shall verify that this process includes instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

7 The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

Findings
PASS
See section 6.3.1.3 for configuration of the cryptographic engine. [AGD] sections “Downloading Software Packages from Juniper Networks” and “Install Junos OS Software Package” describe the update process. See section 6.3.1.4 for details as to what was covered by the EAs.

6.3.2 Preparative Procedures (AGD_PRE.1)

301 The evaluator performs the CEM work units associated with the AGD_PRE.1 SAR. Specific requirements and EAs on the preparative documentation are identified (and where relevant

are captured in the Guidance Documentation portions of the EAs) in the individual EAs for each SFR.

302 Preparative procedures are distributed to Security Administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that Security Administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

303 In addition, the evaluator performs the EAs specified below.

6.3.2.1 Evaluation Activity

304 *The evaluator shall examine the Preparative procedures to ensure they include a description of how the Security Administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target).*

305 The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE product itself).

Findings
PASS
Please refer to work unit AGD_OPE.1-6 in the Evaluation Technical Report [ETR].

6.3.2.2 Evaluation Activity

306 *The evaluator shall examine the preparative procedures to ensure they are provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.*

Findings
PASS
There is only one operational environment claimed in the [ST]. All TOE platforms claimed in the [ST] are covered by the operational guidance. This is evidenced by the platform equivalency.

6.3.2.3 Evaluation Activity

307 *The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.*

Findings
PASS
See previous work unit.

6.3.2.4 Evaluation Activity

308 *The evaluator shall examine the preparative procedures to ensure they include instructions on how to manage the TSF as a product and as a component of the larger Operational Environment in a manner that allows to preserve integrity of the TSF.*

309 *The intent of this requirement is to ensure there exists adequate preparative procedures (guidance in most cases) to put the TSF in a secure state (i.e., evaluated configuration). AGD_PRE.1 lists general requirements, the specific assurance activities implementing it are performed as part of FMT_SMF.1, FMT_MTD.1 and FMT_MOF.1 series of SFRs.*

Findings
PASS
The guidance documentation provides extensive information on managing the security of the TOE as an individual product. Additional best practice guidance provided within those documents help instill a culture of secure manageability within a larger operational environment.

6.3.2.5 Evaluation Activity

310 In addition the evaluator shall ensure that the following requirements are also met.

311 The preparative procedures must:

8 include instructions to provide a protected administrative capability; and

9 identify TOE passwords that have default values associated with them and mandate that they shall be changed.

Findings
PASS
The entire [AGD] document is designed to ensure the administrator is aware of how to configure the TOE to provide a protected administrative capability. No default TOE passwords were identified.

6.4 ALC: Life-cycle Support

6.4.1 Labelling of the TOE (ALC_CMC.1)

312 When evaluating that the TOE has been provided and is labelled with a unique reference, the evaluator performs the work units as presented in the CEM.

Findings
PASS
Please refer to ALC_CMC.1 work units in the Evaluation Technical Report [ETR].

6.4.2 TOE CM Coverage (ALC_CMS.1)

313 When evaluating the developer's coverage of the TOE in their CM system, the evaluator performs the work units as presented in the CEM.

Findings
PASS
Please refer to ALC_CMS.1 work units in the Evaluation Technical Report [ETR].

6.4.3 Systematic Flaw Remediation (ALC_FLR.3) (optional)

314 When evaluating the developer's procedures regarding systematic flaw remediation, the evaluator performs the work units as presented in the CEM.

Findings
PASS
Please refer to ALC_FLR.3 work units in the Evaluation Technical Report [ETR].

6.5 ATE: Tests

6.5.1 Independent Testing – Conformance (ATE_IND.1)

315 The focus of the testing is to confirm that the requirements specified in the SFRs are being met. Additionally, testing is performed to confirm the functionality described in the TSS, as well as the dependencies on the Operational guidance documentation is accurate.

316 The evaluator performs the CEM work units associated with the ATE_IND.1 SAR. Specific testing requirements and EAs are captured for each SFR in [ND-SD] Sections 2, 3 and 4.

317 The evaluator shall consult [ND-SD] Appendix B when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

318 Note that additional Evaluation Activities relating to evaluator testing in the case of a distributed TOE are defined in [ND-SD] Appendix B.4.3.1.

Findings
PASS
The evaluator tested the SFRs by performing the required Test Evaluation Activities for each SFR. The evaluator confirmed the TOE functioned as described in the TSS and the operational guidance was accurate. The ETR covers the ATE_IND.1 CEM work units. The DTR documents the testing strategy and equivalency argument. The TOE is not a distributed TOE.

6.6 Vulnerability Assessment

6.6.1 Vulnerability Survey (AVA_VAN.1)

319 While vulnerability analysis is inherently a subjective activity, a minimum level of analysis can be defined and some measure of objectivity and repeatability (or at least comparability) can be imposed on the vulnerability analysis process. In order to achieve such objectivity and repeatability it is important that the evaluator shall follow a set of well-defined activities and documents their findings so others can follow their arguments and come to the same conclusions as the evaluator. While this does not guarantee that different evaluation facilities will identify exactly the same type of vulnerabilities or come to exactly the same conclusions, the approach defines the minimum level of analysis and the scope of that analysis and provides Certification Bodies a measure of assurance that the minimum level of analysis is being performed by the evaluation facilities.

320 In order to meet these goals some refinement of the AVA_VAN.1 CEM work units is needed. The following table indicates, for each work unit in AVA_VAN.1, whether the CEM work unit is to be performed as written, or if it has been clarified by an Evaluation Activity. If clarification has been provided, a reference to this clarification is provided in the table.

321 Because of the level of detail required for the evaluation activities, the bulk of the instructions are contained in [ND-SD] Appendix A, while an “outline” of the assurance activity is provided below.

6.6.1.1 Evaluation Activity (Documentation)

322 In addition to the activities specified by the CEM in accordance with [ND-SD] Table 2, the evaluator shall perform the following activities.

323 *The evaluator shall examine the documentation outlined below provided by the developer to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously.*

324 The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components should identify compute-capable hardware components, at a minimum that must include the processor, and where applicable, discrete crypto ASICs, TPMs, etc. used by the TOE. Software components include applications, the operating system and other major components that are independently identifiable and reusable (outside of the TOE), for example a web server, protocol or cryptographic implementations, (independently identifiable and reusable components are not limited to the list provided in the example). This additional documentation is merely a list of the name and version number of the components and will be used by the evaluators in formulating vulnerability hypotheses during their analysis.

Findings
PASS
The evaluator collected this information from the developer which was used to feed into the Type 1 Flaw Hypotheses search (below).

325 If the TOE is a distributed TOE then the developer shall provide:

10 documentation describing the allocation of requirements between distributed TOE components as in [NDcPP, 3.4]

11 a mapping of the auditable events recorded by each distributed TOE component as in [NDcPP, Table 2]

12 additional information in the Preparative Procedures as identified in the refinement of AGD_PRE.1 in additional information in the Preparative Procedures as identified in 3.4.1.2 and 3.5.1.2.

Findings
PASS
The TOE is not a distributed TOE.

6.6.1.2 Evaluation Activity

326 The evaluator shall formulate hypotheses in accordance with process defined in [ND-SD] Appendix A. The evaluator shall document the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in [ND-SD] Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with [ND-SD] Appendix A.2. The results of the analysis shall be documented in the report according to [ND-SD] Appendix A.3.

Findings

PASS

The following sources of public vulnerabilities were considered in formulating the specific list of flaws to be investigated by the evaluators, as well as to reference in directing the evaluators to perform key-word searches during the evaluation of the TOE. Hypothesis sources for public vulnerabilities were:

- Juniper Networks, Inc. Security Advisories (https://supportportal.juniper.net/s/global-search/)
- CVEs
    - NIST National Vulnerabilities Database (can be used to access CVE and US-CERT databases identified below): https://web.nvd.nist.gov/view/vuln/search
    - Common Vulnerabilities and Exposures: http://cve.mitre.org/cve/
    - Common Vulnerabilities and Exposures: https://www.cvedetails.com/vulnerability-search.php
- CISA - Known Exploited Vulnerabilities Catalog: https://www.cisa.gov/known-exploited-vulnerabilities-catalog

Type 1 Hypothesis searches were conducted on May 27, 2026 and included the following search terms:

- Junos 24.2R2;
- TOE Models;
    - EX4400-24X
    - EX4400-24T
    - EX4400-24P
    - EX4400-24MP
    - EX4400-48T
    - EX4400-48P
    - EX4400-48F
    - EX4400-48MP
- Processors and PHYs used by the TOE:
    - Intel Atom C3558R

- BCM54192
- BCM54195
- BCM82398
- BCM82399
- BCM82756
- BCM84894M
- BCM54998EM

- OpenSSH;
- OpenSSL;
- FreeBSD

Note: Additional proprietary search terms were also included.

The evaluation team determined that no residual vulnerabilities, that are exploitable by attackers with Basic Attack Potential, exist based on these searches.

No Type 2 flaw hypotheses applied to the TOE based on [ND-SD] sections A.1.2 and A.5.

The evaluation team developed Type 3 flaw hypotheses in accordance with [ND-SD] sections A.1.3 and A.2, and no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.

The evaluation team developed Type 4 flaw hypotheses in accordance with [ND-SD] sections A.1.4 and A.2, and no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.

7 Evaluation Activities for Mandatory SFRs (PKG_SSH)

7.1 Cryptographic Support (FCS)

7.1.1 FCS_SSH_EXT.1 SSH Protocol

7.1.1.1 FCS_SSH_EXT.1.1

7.1.1.1.1 TSS

327 The evaluator shall ensure that the selections indicated in the ST are consistent with selections in this and subsequent components. Otherwise, this SFR is evaluated by activities for other SFRs.

Findings
PASS
The evaluator reviewed [ST] section 6.2.3 and determined that the selections indicated in the ST are consistent with the selections in this and subsequent components.

7.1.1.1.2 Guidance

328 There are no guidance evaluation activities for this component. This SFR is evaluated by activities for other SFRs.

Note	There are no test evaluation activities for this component.

7.1.1.1.3 Tests

329 There are no test evaluation activities for this component. This SFR is evaluated by activities for other SFRs.

Note	There are no test evaluation activities for this component.

7.1.1.2 FCS_SSH_EXT.1.2

7.1.1.2.1 TSS

330 The evaluator shall check to ensure that the authentication methods listed in the TSS are identical to those listed in this SFR component; and, ensure if password-based authentication methods have been selected in the ST then these are also described; and, ensure that if keyboard-interactive is selected, it describes the multifactor authentication mechanisms provided by the TOE.

Findings
PASS

[ST] section 6.2.3 describes all public key authentication methods as well as password-based authentication methods. The evaluator confirmed that the authentication methods listed are identical to those listed in the SFR component.

7.1.1.2.2 Guidance

331 The evaluator shall check the guidance documentation to ensure the configuration options, if any, for authentication mechanisms provided by the TOE are described.

Findings

PASS

[AGD] section "Username and Public Key Authentication over SSH" describes that the TOE supports ECDSA (P-256,P-384, and P-521) and RSA (2048-bit or higher) public keys used for authentication by default.

7.1.1.2.3 Tests

- Test 1: [conditional] If the TOE is acting as SSH Server:

  a. The evaluator shall use a suitable SSH Client to connect to the TOE, enable debug messages in the SSH Client, and examine the debug messages to determine that only the configured authentication methods for the TOE were offered by the server.

High-Level Test Description

Establish an SSH connection to the TOE and enable debug messages in the SSH Client. Confirm that only the configured authentication methods for the TOE were offered by the server.

Findings

PASS - The evaluator has confirmed that only the configured authentication methods for the TOE were offered by the server.

  b. [conditional] If the SSH server supports X509 based Client authentication options:

    a. The evaluator shall initiate an SSH session from a client where the username is associated with the X509 certificate. The evaluator shall verify the session is successfully established.

    b. Next the evaluator shall use the same X509 certificate as above but include a username not associated with the certificate. The evaluator shall verify that the session does not establish.

    c. Finally, the evaluator shall use the correct username (from step a above) but use a different X509 certificate which is not associated with the username. The evaluator shall verify that the session does not establish.

Test Not Applicable TOE does not support X509 based Client authentication.

- Test 2: [conditional] If the TOE is acting as SSH Client, the evaluator shall test for a successful configuration setting of each authentication method as follows:

a. The evaluator shall initiate a SSH session using the authentication method configured and verify that the session is successfully established.

b. Next, the evaluator shall use bad authentication data (e.g. incorrectly generated certificate or incorrect password) and ensure that the connection is rejected.

Steps a-b shall be repeated for each independently configurable authentication method supported by the server.

Test Not Applicable	TOE does not act as an SSH Client

- Test 3: [conditional] If the TOE is acting as SSH Client, the evaluator shall verify that the connection fails upon configuration mismatch as follows:

  a. The evaluator shall configure the Client with an authentication method not supported by the Server.

  b. The evaluator shall verify that the connection fails.

332 If the Client supports only one authentication method, the evaluator can test this failure of connection by configuring the Server with an authentication method not supported by the Client. In order to facilitate this test, it is acceptable for the evaluator to configure an authentication method that is outside of the selections in the SFR.

Test Not Applicable	TOE does not act as an SSH Client

7.1.1.3 FCS_SSH_EXT.1.3

7.1.1.3.1 TSS

333 The evaluator shall check that the TSS describes how “large packets” are detected and handled.

Findings
PASS
[AGD] section 6.2.3 states, “The TOE drops all packets greater than 256KB in an SSH transport connection per RFC 4253.”

7.1.1.3.2 Tests

- Test 1: The evaluator shall demonstrate that the TOE accepts the maximum allowed packet size.

High-Level Test Description
Transmit a packet at maximum capacity allowed by the TOE SSH implementation and verify that the TOE accepts the packet.
Findings
PASS - The evaluator has confirmed the TOE successfully admitted packets at the maximum capacity.

NIAP TD0732

- Test 2: This test is performed to verify that the TOE drops packets that are larger than size specified in the component.

  a. The evaluator shall establish a successful SSH connection with the peer.

  b. Next the evaluator shall craft a packet that is slightly larger than the maximum size specified in this component and send it through the established SSH connection to the TOE. The packet should not be greater than the maximum packet size + 16 bytes. If the packet is larger, the evaluator shall justify the need to send a larger packet.

  c. The evaluator shall verify that the packet was dropped by the TOE. The method of verification will vary by the TOE. Examples include reviewing the TOE audit log for a dropped packet audit or observing the TOE terminates the connection.

High-Level Test Description
Transmit a packet larger than allowed by the TOE SSH implementation and verify that the TOE rejects the packet.
Findings
PASS - The evaluator has confirmed the TOE successfully drops packets beyond the maximum capacity.

7.1.1.4 FCS_SSH_EXT.1.4

7.1.1.4.1 TSS

334 The evaluator will check the description of the implementation of SSH in the TSS to ensure the encryption algorithms supported are specified. The evaluator will check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

Findings
PASS
[ST] section 6.2.3 describes the encryption algorithms supported by the TOE. The evaluator confirmed that the encryption algorithms are identical to those listed for this component.

7.1.1.4.2 Guidance

335 The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

7.1.1.4.3 Tests

336 The evaluator shall perform the following tests.

337 If the TOE can be both a client and a server, these tests must be performed for both roles.

- Test 1: The evaluator must ensure that only claimed algorithms and cryptographic primitives are used to establish an SSH connection. To verify this, the evaluator shall establish an SSH connection with a remote endpoint. The evaluator shall capture the traffic exchanged between the TOE and the remote endpoint during protocol negotiation (e.g. using a packet capture tool or information provided by the endpoint, respectively). The evaluator shall verify from the captured traffic that the TOE offers only the algorithms defined in the ST for the TOE for SSH connections. The evaluator shall perform one successful negotiation of an SSH connection and verify that the negotiated algorithms were included in the advertised set. If the evaluator detects that not all algorithms defined in the ST for SSH are advertised by the TOE or the TOE advertises additional algorithms not defined in the ST for SSH, the test shall be regarded as failed.

  The data collected from the connection above shall be used for verification of the advertised hashing and shared secret establishment algorithms in FCS_SSH_EXT.1.5 and FCS_SSH_EXT.1.6 respectively.

High-Level Test Description
Connect to the TOE using each claimed SSH cipher. Verify the TOE only proposes the claimed ciphers.
Findings
PASS – The evaluator confirmed that the TOE successfully established a connection and that the negotiated algorithms were included in the advertised set.

- Test 2: For the connection established in Test 1, the evaluator shall terminate the connection and observe that the TOE terminates the connection.

Note	Test is satisfied as a part of FCS_SSH_EXT.1.4 Test 1.

- Test 3: The evaluator shall configure the remote endpoint to only allow a mechanism that is not included in the ST selection. The evaluator shall attempt to connect to the TOE and observe that the attempt fails.

High-Level Test Description
Attempt to connect to the TOE using an unclaimed algorithm and verify that the attempt fails.
Findings
PASS – The evaluator confirmed the TOE rejects the connection when the remote endpoint provides an unclaimed algorithm.

7.1.1.5 FCS_SSH_EXT.1.5

7.1.1.5.1 TSS

338 The evaluator will check the description of the implementation of SSH in the TSS to ensure the hashing algorithms supported are specified. The evaluator will check the TSS to ensure that the hashing algorithms specified are identical to those listed for this component.

Findings
PASS
[ST] section 6.2.3 describes the hashing algorithms supported by the TOE. The evaluator confirmed that the hashing algorithms specified are identical to those listed for this component.

7.1.1.5.2 Guidance

339 The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

7.1.1.5.3 Tests

- Test 1: The evaluator shall use the test data collected in FCS_SSH_EXT.1.4, Test 1 to verify that appropriate mechanisms are advertised.

High-Level Test Description
Evaluator shall confirm TOE uses proper encryption to protect data in transit from modification, deletion, and insertion.
Findings
PASS – The evaluator confirmed that only the claimed HMAC algorithms are advertised.

- Test 2: The evaluator shall configure an SSH peer to allow only a hashing algorithm that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection and observe that the connection is rejected.

High-Level Test Description
Evaluator shall attempt to establish an SSH connection with an unsupported hashing algorithm and observe that the connection is rejected by the TOE.
Findings
PASS – The evaluator confirmed the TOE successfully rejects connection when presented with an unsupported hashing method.

7.1.1.6 FCS_SSH_EXT.1.6

7.1.1.6.1 TSS

340 The evaluator will check the description of the implementation of SSH in the TSS to ensure the shared secret establishment algorithms supported are specified. The evaluator will check the TSS to ensure that the shared secret establishment algorithms specified are identical to those listed for this component.

Findings
PASS
[ST] section 6.2.3 lists all the shared secret establishment algorithms supported by the TOE. The evaluator confirmed that the shared secret establishment algorithms specified are identical to those listed for this component.

7.1.1.6.2 Guidance

341 The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

7.1.1.6.3 Tests

NIAP TD0967

- Test 1: The evaluator shall use the test data collected in FCS_SSH_EXT.1.4, Test 1 to verify that appropriate mechanisms are advertised.

  Note that it is permissible for the TOE to advertise the "kex-strict-c-v00@openssh.com" or "kex-strict-s-v00@openssh.com" value alongside the other key exchange mechanisms. These values are associated with client and server, respectively.

High-Level Test Description
Evaluator shall verify the TOE successfully establishes a connection with supported Key Exchange Algorithms.
Findings
PASS – The evaluator confirmed the TOE successfully accepts connection when presented with a supported key exchange algorithm.

- Test 2: The evaluator shall configure an SSH peer to allow only a key exchange method that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection and observe that the connection is rejected.

High-Level Test Description
Attempt connection to TOE and verify the TOE rejects connection when presented with an unsupported Key Exchange Algorithm.
Findings
PASS – The evaluator confirmed the TOE rejects connection when presented with an unsupported Key Exchange Algorithm.

NIAP TD0967

- Test 3: The evaluator shall configure the SSH peer to omit the kex-strict value from the kex_algorithms field. The evaluator shall verify that the TOE is able to successfully establish a connection with the SSH peer.

High-Level Test Description
Using a custom LS SSH Client, attempt to establish a connection where the SSH Client omits the kex-strict value from the kex_algorithms field and show the connection succeeds.
Findings
PASS – The evaluator confirmed the TOE successfully establishes a connection when the kex-strict value is omitted.

NIAP TD0967

- Test 4: The evaluator shall configure the SSH peer to include the kex-strict value in the kex_algorithms field. The evaluator shall verify that the TOE is able to successfully establish a connection with the SSH peer.

High-Level Test Description
Using a custom LS SSH Client, attempt a connection to TOE with a peer that includes the kex-strict value in the kex_algorithms field and show the connection succeeds.
Findings
PASS – The evaluator confirmed the TOE successfully establishes a connection when the kex-strict value is included in the kex_algorithms field.

7.1.1.7 FCS_SSH_EXT.1.7

7.1.1.7.1 TSS

342 The evaluator will check the description of the implementation of SSH in the TSS to ensure the KDFs supported are specified. The evaluator will check the TSS to ensure that the KDFs specified are identical to those listed for this component.

Findings
PASS
[ST] section 6.2.3 lists the KDFs supported by the TOE. The evaluator confirmed that the KDFs specified are identical to those listed for this component.

7.1.1.8 FCS_SSH_EXT.1.8

7.1.1.8.1 TSS

343 The evaluator shall check the TSS to ensure that if the TOE enforces connection rekey or termination limits lower than the maximum values that these lower limits are identified.

Findings
PASS
[ST] section 6.2.3 describes the TOE's rekey limits. The TOE does not enforce connection rekey or termination limits lower than the maximum values.

344 In cases where hardware limitation will prevent reaching data transfer threshold in less than one hour, the evaluator shall check the TSS to ensure it contains:

a. An argument describing this hardware-based limitation and

b. Identification of the hardware components that form the basis of such argument.

345 For example, if specific Ethernet Controller or Wi-Fi radio chip is the root cause of such limitation, these subsystems shall be identified.

Note	The ST does not claim a limitation that prevents reaching the data transfer threshold.

7.1.1.8.2 Guidance

346 The evaluator shall check the guidance documentation to ensure that if the connection rekey or termination limits are configurable, it contains instructions to the administrator on how to configure the relevant connection rekey or termination limits for the TOE.

Findings
PASS
[AGD] section "Limiting the Number of User Login Attempts for SSH Sessions" provides instruction to the administrator on how to configure the relevant connection rekey limits for the TOE.

7.1.1.8.3 Tests

347 The test harness needs to be configured so that its connection rekey or termination limits are greater than the limits supported by the TOE -- it is expected that the test harness should not be initiating the connection rekey or termination.

- Test 1: Establish an SSH connection. Wait until the identified connection rekey limit is met. Observed that a connection rekey or termination is initiated. This may require traffic to periodically be sent, or connection keep alive to be set, to ensure that the connection is not closed due to an idle timeout.

High-Level Test Description
Using a custom SSH client, connect to the TOE and trickle data over the channel to avoid disconnection due to idle timeout. Verify that the TOE rekeys before 1 hour is exceeded.

Findings
PASS – The evaluator confirmed the TOE successfully issues a rekey when threshold is met.

- Test 2: Establish an SSH connection. Transmit data from the TOE until the identified connection rekey or termination limit is met. Observe that a connection rekey or termination is initiated.

High-Level Test Description
Using a custom SSH client and script to generate traffic, connect to the TOE and receive slightly more than 1GB of data. Verify the TOE initiates a rekey before 1GB of data has been encrypted or decrypted using a key.
Findings
PASS – The evaluator confirmed that the TOE initiates a SSH rekey after transmitting 1GB of data.

- Test 3: Establish an SSH connection. Send data to the TOE until the identified connection rekey limit or termination is met. Observe that a connection rekey or termination is initiated.

High-Level Test Description
Using a custom SSH client, connect to the TOE and send slightly more than 1GB of data. Verify the TOE initiates a rekey before 1GB of data has been encrypted or decrypted using a key.
Findings
PASS – The evaluator confirmed that the TOE initiates a SSH rekey after receiving 1GB of data.

8 Optional Requirements (PKG_SSH)

8.1 Strictly Optional Requirements

348 This Package does not define any Strictly Optional requirements.

8.2 Objective Requirements

349 This Package does not define any Objective requirements.

8.3 Implementation-based Requirements

350 This Package does not define any Implementation-based requirements.

9 Evaluation Activities for Selection-based Requirements (PKG_SSH)

9.1 Cryptographic Support (FCS)

9.1.1 FCS_SSHS_EXT.1 SSH Protocol – Server

9.1.1.1 FCS_SSHS_EXT.1

9.1.1.1.1 TSS

351 No activities.

9.1.1.1.2 Guidance

352 The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Findings
PASS
[AGD] section “Configuring SSH on the Evaluated Configuration” contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

9.1.1.1.3 Tests

NIAP TD0682

353 The evaluator shall perform the following tests:

NIAP TD0682

354 Test 1: The evaluator shall use a suitable SSH Client to connect to the TOE and examine the list of server host key algorithms in the SSH_MSG_KEXINIT packet sent from the server to the client to determine that only the configured server authentication methods for the TOE were offered by the server.

High-Level Test Description
Using an SSH client, connect to the TOE server using ssh and check if the TOE offers all claimed hostkey algorithms.
Findings
PASS – The evaluator confirmed that only the configured server authentication methods for the TOE were offered by the server.

NIAP TD0682

355 Test 2: The evaluator shall test for a successful configuration setting of each server authentication method as follows. The evaluator shall initiate a SSH session using the authentication method configured and verify that the session is successfully established. Repeat this process for each independently configurable server authentication method supported by the server.

High-Level Test Description
Using an SSH client, connect to the TOE using a claimed host key algorithm. Verify the connection is successful.
Findings
PASS – The evaluator confirmed the TOE successfully establishes a connection using each of the claimed server authentication methods.

NIAP TD0682

356 Test 3: The evaluator shall configure the peer to only allow an authentication mechanism that is not included in the ST selection. The evaluator shall attempt to connect to the TOE and observe that the TOE sends a disconnect message.

High-Level Test Description
Using an SSH client, attempt to connect to the TOE with an unclaimed host key algorithm. Verify that the connection fails.
Findings
PASS - The evaluator confirmed that the TOE rejects the connection when the client attempts to connect using an unclaimed server authentication method.

10 PP-Module for MACsec Ethernet Encryption (MOD_MACSEC)

357 The EAs defined in this section are only applicable in cases where the TOE claims conformance to a PP-Configuration that includes the NDcPP.

10.1 Modified SFRs

358 The PP-Module does not modify any requirements when the NDcPP is the base.

11 TOE SFR Evaluation Activities (MOD_MACSEC)

11.1 Security Audit (FAU)

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

11.1.1.1 FAU_GEN.1/MACSEC

359 The evaluator shall complete the evaluation activity for FAU_GEN.1 as described in the NDcPP for the auditable events defined in the PP-Module in addition to the applicable auditable events that are defined in the NDcPP. The evaluator shall also ensure that the administrative actions defined for this PP-Module are appropriately audited.

11.2 Cryptographic Support (FCS)

11.2.1 FCS_COP.1/CMAC Cryptographic Operation (AES-CMAC Keyed Hash Algorithm)

11.2.1.1 FCS_COP.1/CMAC

11.2.1.2 TSS

360 The evaluator shall examine the TSS to ensure that it specifies the following values used by the AES-CMAC function: key length, hash function used, block size, and output MAC length.

Findings
PASS
[ST] section 6.2.4 specifies the key length, hash function used, block size and output MAC length used by the AES-CMAC function.

11.2.1.3 Guidance

361 There are no guidance evaluation activities (EAs) for this component.

11.2.1.4 Tests

362 The evaluator shall perform the following tests:

- Test 1: **CMAC Generation Test**

  To test the generation capability of AES-CMAC, the evaluator shall provide to the TSF, for each key length-message length-CMAC length tuple (in bytes), a set of eight arbitrary key-plaintext tuples that will result in the generation of a known MAC value when encrypted. The evaluator shall then verify that the correct MAC was generated in each case.

Note	[ST] Table 20 specifies the CAVP certificate demonstrating the TOE correctly implements AES-CMAC.

- Test 2: **CMAC Verification Test**

  To test the verification capability of AES-CMAC, the evaluator shall provide to the TSF, for each key length-message length-CMAC length tuple (in bytes), a set of 20 arbitrary key-MAC tuples that will result in the generation of known messages when verified. The evaluator shall then verify that the correct message was generated in each case.

  The following information should be used by the evaluator to determine the key length-message length-CMAC length tuples that should be tested:

  - Key length: Values will include the following:
    - 16
    - 32
  - Message length: Values will include the following:
    - 0 (optional)
    - Largest value supported by the implementation (no greater than 65536)
    - Two values divisible by 16
    - Two values not divisible by 16
  - CMAC length:
    - Smallest value supported by the implementation (no less than 1)
    - 16
    - Any supported CMAC length between the minimum and maximum values

Note	[ST] Table 20 specifies the CAVP certificate demonstrating the TOE correctly implements AES-CMAC.

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

11.2.2.1 FCS_COP.1/MACSEC

11.2.2.2 TSS

363 The evaluator shall verify that the TSS describes the supported AES modes that are required for this PP-Module in addition to the ones already required by the NDcPP in FCS_COP.1/DataEncryption.

Findings
PASS
[ST] section 6.2.4 states that the TOE supports AES GCM and AES Key Wrap modes with 128- and 256-bit keys.

11.2.2.3 Guidance

364 There are no guidance EAs for this component.

11.2.2.4 Tests

365 The evaluator shall perform testing for AES-GCM as required by the NDcPP in FCS_COP.1/DataEncryption.

Note	AES-GCM findings can be found in FCS_COP.1/DataEncryption.

366 In addition to the tests specified in the NDcPP for other iterations of FCS_COP.1, the evaluator shall perform the following tests:

- Test 3: **KW-AE Test**: To test the authenticated encryption capability of AES key wrap (KW), the evaluator shall provide five sets of 100 messages and keys to the TOE for each key length supported by the TSF. Each set of messages and keys shall correspond to one of five plaintext message lengths (detailed below). The evaluator shall have the TSF encrypt the messages with the associated key. The evaluator shall verify that the correct ciphertext was generated in each case.

Note	[ST] Table 20 specifies the CAVP certificate demonstrating the TOE correctly implements AES KeyWrap.

NIAP TD0728

- Test 4: **KW-AD Test**: To test the authenticated decryption capability of AES KW, the evaluator shall provide five sets of 100 ciphertext values and keys to the TOE for each key length supported by the TSF. Each set of ciphertexts and keys shall correspond to one of five plaintext message lengths (detailed below). For each set of 100 ciphertext values, 20 shall not be authentic (i.e., fail authentication). The evaluator shall have the TSF decrypt the ciphertext messages with the associated key. The evaluator shall then verify the correct plaintext was generated or the failure to authenticate was correctly detected.

The messages in each set for both tests shall be the following lengths:

- two that are non-zero multiples of 128 bits (two semiblock lengths)
- two that are odd multiples of the semiblock length (64 bits)
- the largest supported plaintext length less than or equal to 4096 bits

Note	[ST] Table 20 specifies the CAVP certificate demonstrating the TOE correctly implements AES KeyWrap.

11.2.3 FCS_MACSEC_EXT.1 MACsec

11.2.3.1 FCS_MACSEC_EXT.1 MACsec

11.2.3.2 TSS

NIAP TD0884

367 The evaluator shall examine the TSS to verify that it describes the ability of the TSF to implement MACsec in accordance with IEEE 802.1AE-2018. The evaluator shall also determine that the TSS describes the ability of the TSF to derive SCI values from peer MAC address and port data and to reject traffic that does not have a valid SCI. Finally, the evaluator shall check the TSS for an assertion that only the claimed EtherTypes are accepted by the MACsec interface. Where the SFR has provided additional EtherType support, the TSS must describe the usage of each of those frame types within the context of MKA, MACsec, and MAC control frame support.

Findings

PASS

[ST] section 6.2.4 states that MACsec is implemented in accordance with IEEE 802.1AE-2018. The evaluator confirmed that the TSS describes the ability of the TSF to derive SCI values from peer MAC address and port data, and to reject traffic that does not have a valid SCI. The evaluator confirmed the TSS contained an assertion that only the claimed EtherTypes are accepted by the MACsec interface in the evaluated configuration.

11.2.3.3 Guidance

NIAP TD0884

368 If the TOE requires enabling or disabling specific EtherTypes to operate within the evaluated configuration, the guidance documentation must provide this information to the administrator.

Findings

PASS

No configuration is required to operate within the evaluated configuration.

11.2.3.4 Tests

369 The evaluator shall perform the following tests:

- Test 5: The evaluator shall successfully establish a MACsec channel between the TOE and a MACsec-capable peer in the operational environment and verify that the TSF logs the communications. The evaluator shall capture the traffic between the TOE and the operational environment to determine the SCI that the TOE uses to identify the peer. The evaluator shall then configure a test system to capture traffic between the peer and the TOE to modify the SCI that is used to identify the peer. The evaluator then verifies that the TOE does not reply to this traffic and logs that the traffic was discarded.

High-Level Test Description
Configure a test MACsec peer to be connected to the TOE's physical interface. Enable MACsec on the TOE and show that a session can be established. Use ping to verify that traffic can be received and is acted upon by the TOE. Then using a custom tool on the test peer, alter the SCI in the MACsec frame while pinging the TOE from the test peer. Show that the TOE no longer responds to the test peer pings and that the TOE accounts for the discarded traffic.
Findings
PASS– The evaluator confirmed that the TOE is able to negotiate a MACsec session and send/receive traffic over the encrypted interface. If the SCI is changed, the TOE will drop the traffic and log it via an accounting metric.

NIAP TD0884

- Test 6: The evaluator shall send Ethernet traffic to the TOE's MAC address that iterates through the full range of supported EtherType values (refer to List of Documented EtherTypes) and observes that traffic for all EtherType values is discarded by the TOE except for the traffic which has an EtherType value described in the SFR. Note that there are a large number of EtherType values so the evaluator is encouraged to execute a script that automatically iterates through each value.

High-Level Test Description
Using a custom tool, generate Ethernet packets with the given EtherType header to the TOE MACsec-enabled interface with the exception of those with EtherType 0x888e, 0x88e5. Show that with the exception of 0x8808, the packets are not acted upon. Using the PCAP from FCS_MACSEC_EXT.1 Test 5, show that packets with ethertype 0x888e and 0x88e5 are acted upon. It is important to realize that it is tricky to properly execute this test case since each of the 3500+ test cases would need to have a well-formed body to really be sure that the packet is being discarded because it is being blocked on the port, rather than being blocked because it is ill-formed. Therefore, we also will test a well-defined packet type 0x0806 (ARP) that should be ignored. We use interface counters to determine if the packets are considered "accepted" or "dropped". If the counters increase, the packet is accepted. If the counter is not increased, then the packet was dropped.
Findings
PASS– The evaluator confirmed that the TOE did not respond to any EtherType other than 0x888e, 0x88e5 or 0x8808 types.

11.2.4 FCS_MACSEC_EXT.2 MACsec Integrity and Confidentiality

11.2.4.1 FCS_MACSEC_EXT.2

11.2.4.2 TSS

370 The evaluator shall examine the TSS to verify that it describes the methods that the TOE implements to provide assurance of MACsec integrity. This should include any confidentiality offsets used, the use of an ICV (including the supported length), and ICV generation with the SAK, using the SCI as the most significant bits of the initialization vector (IV) and the 32 least significant bits of the PN as the IV.

Findings
PASS
[ST] section 6.2.4 describes the methods that the TOE implements to provide assurance of MACsec integrity including: • confidentiality offsets of 0, 30 and 50; • Integrity Check Value (ICV) length between 8 and 16 octets; • ICV generation with the SAK • Using SCI as the most significant bits of the 96-bit IV; and • The 32 least significant bits of the IV are the octets of the PN.

11.2.4.3 Guidance

371 If any integrity verifications are configurable, such as any confidentiality offsets used or the mechanism used to derive an ICK, the evaluator shall verify that instructions for performing these functions are documented.

Findings
PASS
[AGD] section “Configuring MACsec on a Device Running Junos OS” provides instructions for configuring confidentiality offsets.

11.2.4.4 Tests

372 The evaluator shall perform the following tests:

- Test 7: The evaluator shall transmit MACsec traffic to the TOE from a MACsec-capable peer in the operational environment. The evaluator shall verify via packet captures, audit logs, or both that the frame bytes after the MACsec Tag values in the received traffic is not obviously predictable.

High-Level Test Description
Establish a MACsec session and then transmit a large number of ping packets encapsulated within MACsec. Examine the encrypted bytes, including the ICV, and determine that the packet data has high entropy.
Findings
PASS – The evaluator confirmed using quantitative measures that the MACsec payloads are not obviously predictable.

- Test 8: The evaluator shall transmit valid MACsec traffic to the TOE from a MACsec-capable peer in the operational environment that is routed through a test system set up as a man-in-the-middle. The evaluator shall use the test system to intercept this traffic to modify one bit in a packet payload before retransmitting to the TOE. The evaluator shall verify that the traffic is discarded due to an integrity failure.

High-Level Test Description
Configure a test MACsec peer to be connected to the TOE's physical interface. Enable MACsec on the TOE and show that a session can be established. Use ping to verify that traffic can be received and is acted upon by the TOE. Then using a custom tool on the test peer, alter one bit in a MACsec frame while pinging the TOE from the test peer. Show that the TOE no longer responds to the test peer pings and that the TOE accounts for the discarded traffic.
Findings
PASS – The evaluator confirmed that when MACsec packet bodies are modified, the TOE will discard the packets.

11.2.5 FCS_MACSEC_EXT.3 MACsec Randomness

11.2.5.1 FCS_MACSEC_EXT.3

11.2.5.2 TSS

373 The evaluator shall examine the TSS to verify that it describes the method used to generate SAKs and nonces and that the strength of the CAK and the size of the CAK's key space are provided.

Findings
PASS
[ST] section 6.2.4 describes that the SAKs are generated using KDF function AES-CMAC-128 or AES-CMAC-256 based on the configured cipher suite. The strength of the CAK and size of the CAK's key space can be 128 bits or 256 bits.
Nonces are generated from an RNG each time a SAK is generated.

11.2.5.3 Guidance

374 There are no guidance EAs for this component.

11.2.5.4 Tests

375 Testing of the TOE's MACsec capabilities and verification of the deterministic random bit generator is sufficient to demonstrate that this SFR has been satisfied.

11.2.6 FCS_MACSEC_EXT.4 MACsec Key Usage

11.2.6.1 FCS_MACSEC_EXT.4

11.2.6.2 TSS

376 The evaluator shall check the TSS to ensure that it describes how the SAK is wrapped prior to being distributed using the AES implementation specified in this PP-Module.

Findings
PASS
[ST] section 6.2.4 describes that each distributed SAK is protected by AES Key Wrap method when transmitted.

11.2.6.3 Guidance

377 If the method of peer authentication is configurable, the evaluator shall verify that the guidance provides instructions on how to configure this. The evaluator shall also verify that the method of specifying a lifetime for CAKs is described.

Findings
PASS
[AGD] section “Configuring MACsec on a Device Running Junos OS” provides instructions on configuring the PSK authentication. No other authentication method is claimed.
[AGD] sections “Configuring MACsec with keychain using Layer 3 Traffic” and “Configuring MACsec with keychain using Layer 2 Traffic” describes the method of specifying a lifetime for CAKs.

11.2.6.4 Tests

378 The evaluator shall perform the following tests:

- Test 9: For each supported method of peer authentication in FCS_MACSEC_EXT.4.1, the evaluator shall follow the operational guidance to configure the supported method (if applicable). The evaluator shall set up a packet sniffer between the TOE and a MACsec-capable peer in the operational environment. The evaluator shall then initiate a connection between the TOE and the peer such that authentication occurs and a secure connection is established. The evaluator shall wait one minute and then disconnect the TOE from the peer and stop the sniffer. The evaluator shall use the packet captures to verify that the SC was established via the selected mechanism and that the non-VLAN EtherType of the first data frame sent between the TOE and the peer is 88-E5.

High-Level Test Description
Establish a MACsec session and then transmit 60 ping packets, 1 per-second. Terminate the MACsec session and show that the only packets on the wire are those with EtherType 0x888e (EAPOL) and 0x88e5 (MACsec). Specifically show that a MACsec packet appears only after the EAPOL packets have negotiated a CA.
Findings

PASS – The evaluator confirmed that the TOE is capable of negotiating a MACsec session and that the only packets on the wire are either MKA EAPOL packets (0x888e) or MACsec (0x88e5) and that specifically, MACsec packets only appear after a secure CA has been established.

- Test 10: The evaluator shall capture traffic between the TOE and a MACsec-capable peer in the operational environment. The evaluator shall then cause the TOE to distribute a SAK to that peer, capture the MKPDUs from that operation, and verify the key is wrapped in the captured MKPDUs.

High-Level Test Description
Establish a MACsec session and review the distribution of the SAK to determine that the TOE has wrapped the SAK using encryption in the MKPDU. NOTE: Since the SAK is derived (indirectly) from the CAK, there is no way to prove that the SAK is actually wrapped, except that we use known-good tools that expect the SAK to be wrapped and capable of being decrypted.
Findings
PASS – The evaluator confirmed that the TOE-generated SAK is wrapped in the MKPDU.

11.2.7 FCS_MKA_EXT.1 MACsec Key Agreement

11.2.7.1 FCS_MKA_EXT.1.1

11.2.7.2 FCS_MKA_EXT.1.2

11.2.7.3 FCS_MKA_EXT.1.3

11.2.7.3.1 TSS

379 The evaluator shall examine the TSS to verify that it describes the methods that the TOE implements to provide assurance of MKA integrity, including the use of an ICV and the ability to use a KDF to derive an ICK.

Findings
PASS
[ST] section 6.2.4 states, Each MACsec Key Agreement protocol data unit (MKPDU) transmitted is integrity protected by a 128-bit Integrity Check value (ICV), generated by AES-CMAC using the Integrity Check value Key (ICK). The ICV Key (ICK) is derived from CAK (using AES_CMAC).

11.2.7.3.2 Guidance

380 There are no guidance EAs for this element.

11.2.7.3.3 Tests

381 The evaluator shall perform the following tests:

- Test 11: The evaluator shall transmit MKA traffic (MKPDUs) to the TOE from an MKA-capable peer in the operational environment. The evaluator shall verify via

packet captures, audit logs, or both that the last 16 octets of the MKPDUs in the received traffic do not appear to be predictable.

High-Level Test Description
Establish a MACsec session and leave the connection long enough to capture 7813 MKPDUs originating from the TOE (this will yield 1M bits from the 16-byte ICV). Examine the ICV and determine that the packet data has high entropy.
Findings
PASS – The evaluator confirmed using quantitative measures that the MKPDU ICVs are not obviously predictable.

- Test 12: The evaluator shall transmit valid MKA traffic to the TOE from an MKA-capable peer in the operational environment that is routed through a test system set up as a man-in-the-middle. The evaluator shall use the test system to intercept this traffic to modify one bit in a packet payload before retransmitting to the TOE. The evaluator shall verify that the traffic is discarded due to an integrity failure.

High-Level Test Description
Configure a test MACsec peer to be connected to the TOE's physical interface. Enable MACsec on the TOE. Then using a custom tool on the test peer, alter one bit in an MKPDU while attempting to establish a secure session. Show that the TOE drops the MKPDU frame.
Findings
PASS – The evaluator confirmed that when the ICV bytes are modified, the TOE will drop them and fail to establish a secure session.

11.2.7.4 FCS_MKA_EXT.1.4

11.2.7.4.1 TSS

382 There are no TSS EAs for this element.

11.2.7.4.2 Guidance

383 There are no guidance EAs for this element.

11.2.7.4.3 Tests

NIAP TD0882

384 The tests below require the TOE to be deployed in an environment with two MACsec-capable peers, identified as devices B and C, that the TOE can communicate with. Prior to performing these tests, the evaluator shall follow the steps in the guidance documentation to configure the TOE as the key server and principal actor (peer). The evaluator shall then perform the following tests using a traffic sniffer to capture this traffic:

- Test 13a: The evaluator shall configure the TOE to establish a MKA session with a new peer. The evaluator shall verify that the TOE sends a fresh SAK to the peer and sends other MKPDUs required for a new session. The evaluator shall verify from packet captures that MKPDUs are

sent at least once every two seconds or every half-second, in accordance with the SFR selection.

High-Level Test Description
The TOE is configured to be able to communicate with two independent peers and the MKA session packets are sent every 2 seconds.
Findings
PASS - The evaluator confirmed the TOE is capable of establishing an MKA session using a predefined CAK/CKN and that MKPDUs are distributed every 2 seconds and every 0.5 seconds when MKA Bounded Hello Time limit is configured.

NIAP TD0882

- Test 13b: (Conditional - If "EAPTLS with DevIDs" is selected in FCS_MACSEC_EXT.4.1) The evaluator shall use EAP-TLS to derive a CAK and configure the TOE's peer to send "0" in the MKA parameter field for MACsec Capability (Table 11-6 in 802.1X-2020). The evaluator shall observe that the peer is deleted from the connection after MKA Life Time has passed.

Test Not Applicable: The ST has not selected “EAPTLS with DevIDs”.

NIAP TD0882

- Test 14a: (Conditional - if any "group CAK" selection is made in FCS_MKA_EXT.1.5) The evaluator shall configure the TOE to send a fresh SAK with two peers as active participants. The evaluator shall verify that the TOE sends a fresh SAK to the peers and sends other MKPDUs required for a new session. The evaluator shall verify from packet captures that MKPDUs are sent at least once every two seconds or every half second in accordance with the SFR Selection.

Test Not Applicable: The ST has not selected a “group CAK” selection.

NIAP TD0882

- Test 14b: (Conditional – if any “group CAK” selection is made in FCS_MKA_EXT.1.5) Disconnect one of the peers. Arbitrarily introduce an artificial delay in sending a fresh SAK following the change in the Live Peer List. For this delayed fresh SAK, use a man-in-the-middle device to observe that the MKA Life Time of 6.0 seconds is enforced by the TSF.

Test Not Applicable: The ST has not selected a “group CAK” selection.

11.2.7.5 FCS_MKA_EXT.1.5

11.2.7.6 FCS_MKA_EXT.1.6

11.2.7.7 FCS_MKA_EXT.1.7

11.2.7.7.1 TSS

385 The evaluator shall verify that the TSS describes the TOE's compliance with IEEE 802.1X-2010 and 802.1Xbx-2014 for MKA, including the values for MKA and Hello timeout limits and support for data delay protection. The evaluator shall also verify that the TSS describes the ability of the PAE of the TOE to establish unique CAs with individual peers and group CAs using a group CAK such that a new group SAK is distributed every time the group's membership changes. The evaluator shall also verify that the TSS describes the invalid MKPDUs that are discarded automatically by the TSF in a manner that is consistent with the SFR, and that valid MKPDUs are decoded in a manner consistent with IEEE 802.1X-2010 section 11.11.4.

Findings

PASS

[ST] section 6.2.4 describes the TOE's compliance with IEEE 802.1X-2010 and 802.1Xbx-2014 for MKA, including MKA life time of 6s and MKA Hello Time and MKA Bounded Hello Timeout of 2s and 0.5s respectively. The TSS describes the ability of the PAE of the TOE to establish a unique Connectivity-Association (CA) per physical port (IFD). The TOE does not support group CAs.

[ST] section 6.2.4 describes the invalid MKPDUs that are discarded automatically by the TSF that is consistent with the SFR and that valid MKPDUs are decoded in a manner consistent with IEEE 802.1X-2010 section 11.11.4.

11.2.7.7.2 Guidance

386 The evaluator shall verify that the guidance documentation provides instructions on how to configure the TOE to act as the key server in an environment with multiple MACsec-capable devices.

Findings

PASS

[AGD] section "Configuring MACsec on a Device Running Junos OS" provides instructions on how to configure the TOE to act as a key server in an environment with multiple MACsec capable devices.

11.2.7.7.3 Tests

NIAP TD0882

387 The tests below require the TOE to be deployed in an environment with two MACsec-capable peers, identified as devices B and C, that the TOE can communicate with. Prior to performing these tests, the evaluator shall follow the steps in the guidance documentation to configure the TOE as the key server and principal actor (peer). The evaluator shall then perform the following tests:

- Test 15: (Conditional – if any “group CAK” selection is made in FCS_MKA_EXT.1.5) The evaluator shall perform the following steps:

  1. Load one PSK onto the TOE and device B and a second PSK onto the TOE and device C. This defines two pairwise CAs.

  2. Generate a group CAK for the group of three devices using ieee8021XKayCreateNewGroup.

  3. Observe via packet capture that the TOE distributes the group CAK to the two peers, protected by AES key wrap using their respective PSKs.

  4. Verify that B can form an SA with C and connect securely.

  5. Disable the KaY functionality of device C using ieee8021XPaePortKayMkaEnable.

  6. Generate a group CAK for the TOE and B using ieee8021XKayCreateNewGroup and observe they can connect.

  7. The evaluator shall have B attempt to connect to C and observe this fails.

  8. Re-enable the KaY functionality of device C.

  9. Invoke ieee8021XKayCreateNewGroup again.

  10. Verify that both the TOE can connect to C and that B can connect to C.

Test Not Applicable: The ST has not selected a “group CAK” selection.

NIAP TD0889

- Test 16: The evaluator shall start an MKA session between the TOE and an environmental MACsec peer and then perform the following steps:

  1. Send an MKPDU to the TOE's individual MAC address from a peer. Verify the frame is dropped and logged.

  2. Send an MKPDU to the TOE that is less than 32 octets long. Verify the frame is dropped and logged.

  3. Send an MKPDU to the TOE whose length in octets is not a multiple of four. Verify the frame is dropped and logged.

  4. Send an MKPDU to the TOE that is one byte short. Verify the frame is dropped and logged.

  5. Send an MKPDU to the TOE with unknown Agility Parameter. Verify the frame is dropped and logged.

High-Level Test Description
Using a custom tool, introduce the errors requested within the MKPDUs and show that they do not result in an established connection.

Findings
PASS – The evaluator confirmed that MKPDUs will not be accepted if they are: 1) sent to the direct interface address, 2) less than 32 bytes, 3) not a multiple of four bytes, 4) not long enough, and 5) having an invalid agility parameter.

11.3 Identification and Authentication (FIA)

11.3.1 FIA_PSK_EXT.1 Pre-Shared Key Composition

11.3.1.1 FIA_PSK_EXT.1

11.3.1.2 TSS

388 The evaluator shall examine the TSS to ensure it describes the process by which the bit-based PSKs are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1.

Findings
PASS
[ST] section 6.3.1 states, “The TSF accepts bit-based pre-shared keys entered as a string of up to 64 hexadecimal characters.”
The TOE does not support generating bit-based pre-shared keys.

11.3.1.3 Guidance

389 The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on the composition of strong PSKs, and (if the selection indicates keys of various lengths can be entered) that it provides information on the range of lengths supported.

390 The evaluator shall confirm the operational guidance contains instructions for either entering bit-based PSKs for each protocol identified in the requirement, generating a bit-based PSK, or both.

Findings
PASS
[AGD] section “Overview of Media Access Control Security (MACsec) in FIPS mode” describes that the PSK can be between 32 and 64 digits hexadecimal number for CAKs, and 64 digit hexadecimal number for CKNs. The guidance provides warnings on the composition of strong PSKs.
The operational guidance contains instructions for entering a bit-based PSK for each protocol identified in the requirement.
The guidance does not claim support for generating a bit-based PSK.

11.3.1.4 Tests

391 The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Note that one or more of these tests can be performed with a single test case.

- Test 17: (conditional, the TOE supports PSKs of multiple lengths) The evaluator shall use the minimum length, the maximum length, a length inside the allowable range, and invalid lengths beyond the supported range (both higher and lower). The minimum, maximum, and included length tests should be successful, and the invalid lengths must be rejected by the TOE.

High-Level Test Description
Configure a macsec key to be exactly 16 octets when a 128-bit cipher is used. Show it is accepted.
Configure a macsec key to be 15 octets when a 128-bit cipher is used. Show it is accepted with a warning.
Configure a macsec key to be 17 octets when a 128-bit cipher is used. Show it is rejected.
Configure a macsec key to be 0 octets when a 128-bit cipher is used. Show it is rejected.
Configure a macsec key to be exactly 32 octets when a 256-bit cipher is used. Show it is accepted.
Configure a macsec key to be exactly 16 octets when a 256-bit cipher is used. Show it is accepted with warning.
Configure a macsec key to be exactly 33 octets when a 256-bit cipher is used. Show it is rejected.
Configure a macsec key to be 0 octets when a 256-bit cipher is used. Show it is rejected.
Findings
PASS - – The evaluator confirmed the TOE supports keys of 16 octets for 128-bit ciphers and 32 octets for 256-bit ciphers. Anything lower than the supported length will be padded with zeros as described in the AGD. Any length above the supported length results in the PSK being rejected.

- Test 18: (conditional, the TOE does not generate bit-based PSKs) The evaluator shall obtain a bit-based PSK of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

High-Level Test Description
Configure a valid CAK for a 128-bit cipher and show that the key can be used to negotiate a successful session.
Configure a valid CAK for a 256-bit cipher and show that the key can be used to negotiate a successful session.
Findings
PASS – The evaluator confirmed that both 128-bit and 256-bit PSKs can be used to establish a session.

- Test 19: (conditional, the TOE can generate bit-based PSKs) The evaluator shall generate a bit-based PSK of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

Test not Applicable:	The TOE does not generate bit-based pre-shared keys.

11.4 Security Management (FMT)

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

11.4.1.1 FMT_SMF.1/MACSEC

11.4.1.2 TSS

392 The evaluator shall verify that the TSS describes the ability of the TOE to provide the management functions defined in this SFR.

Findings
PASS
[ST] section 6.4.1 the evaluator verified that the TSS describes the ability of the TOE to provide the management functions defined in this SFR.

11.4.1.3 Guidance

393 The evaluator shall examine the operational guidance to determine that it provides instructions on how to perform each of the management functions defined in this SFR.

Findings
PASS
The evaluator determined that [AGD] section “Configure MACsec” provides instruction on how to perform each of the management functions defined in this SFR.

NIAP TD0803

394 (conditional, the length of the CKN is configurable) The evaluator shall verify that the guidance describes how to set the CKN length to 1-32 octets.

Findings
PASS
[AGD] section “Configure MACsec” describes that the CKN as a 64 digit hexadecimal number (32 octets) and is not configurable.

11.4.1.4 Tests

395 The evaluator shall set up an environment where the TOE can connect to two other MACsec devices, identified as devices B and C, with the ability of PSKs to be distributed between them. The evaluator shall configure the devices so that the TOE will be elected key server and principal actor, i.e., has highest key server priority.

396 The evaluator shall follow the relevant operational guidance to perform the tests listed below. Note that if the TOE claims multiple management interfaces, the tests should be performed for each interface that supports the functions.

NIAP TD0803

- Test 20: The evaluator shall connect to the PAE of the TOE and install a PSK. The evaluator shall then specify a CKN and that the PSK is to be used as a CAK.
    - Repeat this test for both 128-bit and 256-bit key sizes.
    - Repeat this test for a CKN of valid length (1-32 octets), and observe success.
    - (conditional, the length of the CKN is not configurable) Repeat this test again for CKN of invalid lengths zero and 33, and observe failure.
    - (conditional, the length of the CKN is configurable) The evaluator shall observe that the admin guidance instructs the admin to configure the TOE to only allow CKN lengths of 1-32 bytes.

High-Level Test Description
Configure a 128-bit key with a CKN of a valid length. Show that a MACsec session can be established. Configure a 256-bit key with a CKN of a valid length. Show that a MACsec session can be established. Attempt to configure a CKN of length 0 octets. Show that the TOE rejects the CKN. Attempt to configure a CKN of length 33 octets. Show that the TOE rejects the CKN.
Findings
PASS – The evaluator confirmed that CKNs with valid sizes permit successful negotiations and CKNs with invalid sizes cannot be set on the TOE.

NIAP TD0889

- Test 21: (conditional, "Cause key server to generate a new group CAK..." is selected) The evaluator shall test the ability of the TOE to enable and disable MKA participants using the management function specified in the ST. The evaluator shall install PSKs in devices B and C, and take any necessary additional steps to create corresponding MKA participants. The evaluator shall disable the MKA participant on device C, then observe that the TOE can communicate with B but cannot communicate with device C. The evaluator shall re-enable the MKA participant of device C and observe that the TOE is now able to communicate with devices B and C.

Test Not Applicable:	“Cause key server to generate a new group CAK” is not selected in the ST.

- Test 22: For TOEs using only PSKs, the TOE should be the key server in both tests and only one peer (B) needs to be tested. The tests are:

NIAP TD0840

- Test 22.1: (Conditional: The TOE supports MKA keychains with multiple CKN/CAKs): Switch to unexpired CKN: TOE and Peer B have CKN1(10 minutes) and CKN2. CKN2 can either be configured with a longer overlapping lifetime (20 minutes) or be configured with a lifetime starting period of more than 10 minutes after the CKN1 start. The TOE and Peer B start using CKN1 and after 10 minutes, verify that the TOE expires SAK1. This can be verified by either 1) seeing the TOE immediately distribute a new SAK to the peer if

the lifetime of CKN2 overlaps CKN1, or 2) by terminating the connection with CKN1 and distributing a new SAK once the lifetime period of CKN2 begins.

High-Level Test Description
Configure two CKNs: bb01 will have an expiry 10 minutes from the start of the test; bb02 will be valid for 20 minutes from the start of the test, thereby overlapping with bb02. Establish a session between the TOE and the peer and show that CKN bb01 expires after 10 minutes and CKN bb02 is capable of being used to provide uptime after bb01 expires.
Findings
PASS – The evaluator confirmed that the TOE is capable of operating a key schedule. Specifically, after a key expires, if there is another key ready to be used, the TOE will make use of the key to generate a new SAK.

- Test 22.2: Reject CA with expired CKN: TOE has CKN1 (10 minutes). Peer B has CKN1 (20 minutes). TOE and Peer B start using CKN1 and after 10 minutes, verify that the TOE rejects (or ignores) peer's request to use (or distribute) a SAK using CKN1.

High-Level Test Description
Using the configuration from test 22.1, configure a single CKN on the TOE set to start 10 minutes from the start of the test. Establish a session between the TOE and node B using the currently active bb02 from test 22.1. Show that bb02 expires after 10 minutes when bb03 becomes active, and is no longer used even if the test tool continues to advertise support.
Findings
PASS – The evaluator confirmed that when the TOE expires a key, peers can no longer continue to make use of the expired CKN to form a secure CA.

- Test 23: (conditional, "Cause key server to generate a new group CAK..." is selected) The evaluator shall connect to the PAE of the TOE, set the management function specified in the ST (e.g., set ieee8021XKayCreateNewGroup to true), and observe that the TOE distributes a new group CAK.

Test Not Applicable: "Cause key server to generate a new group CAK" is not selected in the ST.

11.5 Protection of the TSF (FPT)

11.5.1 FPT_CAK_EXT.1 Protection of CAK Data

11.5.1.1 FPT_CAK_EXT.1

11.5.1.2 TSS

397 The evaluator shall examine the TSS to determine that it details how CAKs are stored and that they are unable to be viewed through an interface designed specifically for that purpose. If these values are not stored in plaintext, the TSS shall describe how they are protected or obscured.

Findings

PASS

[ST] section 6.5.4 states that the CAKs are stored obfuscated and they are unable to be viewed through an interface designed specifically for that purpose.

11.5.1.3 Guidance

398 There are no guidance EAs for this component.

11.5.1.4 Tests

399 There are no test EAs for this component.

11.5.2 FPT_FLS.1 Failure with Preservation of Secure State

11.5.2.1 FPT_FLS.1

11.5.2.2 TSS

NIAP TD0816

400 The evaluator shall examine the TSS to determine that it indicates that the TSF will attain a secure/safe state (e.g., shutdown) if a self-test failure is detected. For TOEs with redundant failover capability, the evaluator shall examine the TSS to determine that it indicates that the failed components will attain a secure/safe state (e.g., shutdown) if a self-test failure is detected.

Findings

PASS

[ST] section 6.5.2 describes that in the event of a failure of any of the self-tests, the TOE will enter a FIPS error state (panic) and reboot. No command line input or traffic to any interface is processed during this time.

The TOE does not have redundant failover capabilities.

11.5.2.3 Guidance

401 The evaluator shall examine the operational guidance to verify that it describes the behavior of the TOE following a self-test failure and actions that an administrator should take if it occurs.

Findings

PASS

[AGD] section “Understanding FIPS Self-Tests” the TOE enters FIPS error state and reboots.

11.5.2.4 Tests

NIAP TD0816

402 The following test may require the vendor to provide access to a test platform that provides the evaluator with the ability to modify the TOE internals in a manner that is not provided to end customers:

- Test 24: For each failure mode specified in the ST that can be deliberately induced, the evaluator shall ensure that the TOE attains a secure state (e.g., shutdown) after initiating each failure mode type. For TOEs with redundant failover capability, the evaluator shall determine that the failed components attain a secure state and the behavior of the TOE is consistent with the operational guidance. For each component, the evaluator shall repeat each type of self-test that can be deliberately induced to fail.

High-Level Test Description
Load a debug image onto the TOE which is designed to deliberately induce KAT self-test failures. Ensure that the TOE attains a secure state after initiating each failure mode type.
Findings
PASS – The evaluator confirmed the TOE attains a secure state and reboots after initiating each failure mode type.

11.5.3 FPT_RPL.1 Replay Detection

11.5.3.1 FPT_RPL.1

11.5.3.2 TSS

NIAP TD0881

403 The evaluator shall examine the TSS to determine that it describes how replay is detected for MKPDUs and MPDUs and how replayed MKPDUs and MPDUs are handled by the TSF.

Findings
PASS
[ST] section 6.5.4 describes how replay is detected for MKPDUs and MPDUs. Replayed data from MPDUs and MKPDUs will be discarded and logged by the TOE.

11.5.3.3 Guidance

404 There are no guidance EAs for this component.

11.5.3.4 Tests

405 The evaluator shall perform the following tests:

NIAP TD0881

406 Before performing each test, the evaluator shall successfully establish a MACsec channel between the TOE and a MACsec-capable peer in the operational environment sending enough traffic to see it working and verify the MN and PN values increase for each direction.

NIAP TD0746

- Test 25: The evaluator shall set up a MACsec connection with an entity in the operational environment. The evaluator shall then capture traffic sent from this remote entity to the TOE. The evaluator shall retransmit copies of this traffic to the TOE in order to impersonate the remote entity where the PN values in the SecTag of these packets are less than the lowest acceptable PN for the SA. The evaluator shall observe that the TSF does not take action in response to receiving these packets and that the audit log indicates that the replayed traffic was discarded.

High-Level Test Description
Establish a MACsec connection and ensure that the TOE can send/receive ping packets. Capture an incoming ping packet with a packet number (PN) that is less than the current replay windows and re-transmit the packet contents to the TOE from the test tool. The TOE will ignore the packet.
Findings
PASS – The evaluator confirmed that MACsec packets that fall outside of the replay window are rejected and captured in the packet auditing metrics.

NIAP TD0881

- Test 26: The evaluator shall capture frames during an MKA session and record the lowest MN from the Basic Parameter set observed in a particular time range. The evaluator shall then send a frame with a lower MN, and then verify that this frame is dropped. The evaluator shall verify that the device logged this event.

High-Level Test Description
Using a custom tool, capture MKPDUs delivered to the TOE and replay one of them. The replay will be dropped.
Findings
PASS – The evaluator confirmed that when an MKPDU MN is replayed, the TOE will drop reject the MKPDU and the failure is audited.

11.6 Trusted Path/Channels (FTP)

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

11.6.1.1 FTP_ITC.1/MACSEC

407 This SFR is addressed through evaluation of FCS_MACSEC_EXT.1 through FCS_MACSEC_EXT.4.

12 Evaluation Activities for Optional SFRs (MOD_MACSEC)

408 No optional SFRs were selected by the TOE.

13 Evaluation Activities for Selection-Based SFRs (MOD_MACSEC)

409 No selection-based SFRs were selected by the TOE.

14 Evaluation Activities for Objective SFRs (MOD_MACSEC)

410 The PP-Module does not define any objective requirements.

15 Evaluation Activities for Implementation-based SFRs (MOD_MACSEC)

411 The PP-Module does not define any implementation-based requirements.

16 Evaluation Activities for SARs (MOD_MACSEC)

412 The PP-Module does not define any SARs beyond those defined within the base NDcPP to which it must claim conformance. It is important to note that a TOE that is evaluated against the PP-Module is inherently evaluated against this Base-PP as well. The NDcPP includes a number of Evaluation Activities associated with both SFRs and SARs. Additionally, the PP-Module includes a number of SFR-based Evaluation Activities that similarly refine the SARs of the Base-PPs. The evaluation laboratory will evaluate the TOE against the Base-PP and supplement that evaluation with the necessary SFRs that are taken from the PP-Module.