Senetas

Distributed by Thales

CN Series Encryptors 5.5.0

Assurance Activity Report

Version 1.2

December 2024

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Author	Reviewer	Description
0.1	6/5/2024	J. Sim		Initial Draft
0.2	10/2/2024	J. Sim	C. Cantlon	Release to QA
1.0	10/23/2024	J. Sim	C. Cantlon	Added TD0886 Finalizing the document
1.1	12/16/2024	J. Sim		Addressing ECR comments
1.2	12/16/2024	J. Sim		Addressing CAVP comments

Table of Contents

1 INTRODUCTION 4
1.1 Evaluation Identifiers 4
1.2 Evaluation Methods 4
1.3 Reference Documents 5

2 TESTING OVERVIEW 7
2.1 TOE Components 7
2.2 Version Verification 8
2.3 Non-TOE Components 9
2.4 Test Environment 9
2.5 Test Platform Equivalency 11

3 EVALUATION ACTIVITIES FOR MANDATORY SFRS (NDCPPV3) 14
3.1 Security Audit (FAU) 14
3.2 Cryptographic Support (FCS) 21
3.3 Identification and Authentication (FIA) 41
3.4 Security management (FMT) 43
3.5 Protection of the TSF (FPT) 48
3.6 TOE Access (FTA) 55
3.7 Trusted path/channels (FTP) 58

4 EVALUATION ACTIVITIES FOR OPTIONAL REQUIREMENTS (NDCPPV3) 62

5 EVALUATION ACTIVITIES FOR SELECTION-BASED REQUIREMENTS (NDCPPV3) 63
5.1 Identification and Authentication (FIA) 63
5.2 Protection of the TSF (FPT) 65
5.3 Security management (FMT) 65
5.4 TOE Access (FTA) 71

6 EVALUATION ACTIVITIES FOR MANDATORY SFRS (PKG_SSH) 73
6.1 Cryptographic Support (FCS) 73

7 EVALUATION ACTIVITIES FOR OPTIONAL SFRS (PKG_SSH) 83
7.1 Strictly Optional Requirements 83
7.2 Objective Requirements 83
7.3 Implementation-based Requirements 83

8 EVALUATION ACTIVITIES FOR SELECTION-BASED SFRS (PKG_SSH) 84
8.1 Cryptographic Support (FCS) 84

9 EVALUATION ACTIVITIES FOR SECURITY ASSURANCE REQUIREMENTS 86
9.1 ASE: Security Target Evaluation 86
9.2 ADV: Development 87
9.3 AGD: Guidance Documents 88
9.4 ALC: Life-cycle Support 92
9.5 ATE: Tests 93
9.6 Vulnerability Assessment 94
9.7 Evaluating Additional Components for a Distributed TOE 96

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	Senetas Corporation Ltd, Distributed by Thales SA (SafeNet)
TOE	Senetas Distributed by Thales CN Series Encryptors 5.5.0 Build: 31224
Security Target	Senetas Distributed by Thales CN Series Encryptors 5.5.0 Security Target, v1.7, December 2024
Protection Profile	collaborative Protection Profile for Network Devices, Version: 3.0e, Date: 06-December-2023 [NDcPP] Functional Package for Secure Shell (SSH), Version: 1.0, Date 13-May-2021 [PKG_SSH]

1.2 Evaluation Methods

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

Table 2: Evaluation Methods

Evaluation Criteria	CC v3.1R5
Evaluation Methodology	CEM v3.1R5
Supporting Documents	Evaluation Activities for Network Device cPP, Version 3.0e, Date: 06-December-2023 [ND-SD]

Table 3: Technical Decisions

NDcPP v3.0e Technical Decisions	Applicable
TD0836: NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1	Applicable

NDcPP v3.0e Technical Decisions	Applicable
TD0868: NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	Not Applicable FCS_IPSEC_EXT.1 not claimed
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
TD0886: Clarification to FAU_STG_EXT.1 Test 6	Applicable

SSH Functional Package v1.0 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
TD0732: FCS_SSHS_EXT.1.3 Test 2 Update	Applicable
TD0777: Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Applicable

1.3 Reference Documents

Table 4: List of Reference Documents

Ref	Document
[ST]	Senetas Distributed by Thales CN Series Encryptors 5.5.0 Security Target, v1.7, December 2024
[AGD]	Senetas Distributed by Thales CN4000/CN6000/CN9000 Series Ethernet Encryptors Firmware Version 5.5.0 Operational User Guidance (AGD_OPE.1), v1.1, 16 December 2024 Senetas Corporation CN4010 Encryptor All Operational Modes, Rev 55-24-010, October 2024 Senetas Corporation CN4020 Encryptor All Operational Modes, Rev 55-24-010, October 2024 Senetas Corporation CN6010 Encryptor All Operational Modes, Rev 55-24-010, October 2024

Ref	Document
	Senetas Corporation CN6110 Encryptor All Operational Modes, Rev 55-24-010, October 2024 Senetas Corporation CN6140 Encryptor All Operational Modes, Rev 55-24-010, October 2024 Senetas Corporation CN9120 Encryptor Ethernet Mode, Rev 55-24-010, October 2024
[NDcPP]	Collaborative Protection Profile for Network Devices, Version: 3.0e, 06 December 2023
[ND-SD]	Evaluation Activities for Network Device cPP, Version: 3.0e, 06 December 2023
[PKG_SSH]	Functional Package for Secure Shell (SSH), Version 1.0, 13 May 2021

2 Testing Overview

3 Testing was performed by Joon Sim from June 2024 to December 2024. Testing was performed in the Lightship Baltimore facility that has been accredited by NVLAP. The TOE and test setup was physically and logically protected from unauthorized access, so the integrity TOE and testing results can be assured.

2.1 TOE Components

Model	CPU & ASIC	Hardware	Power	Protocol / FPGA Bitstream	AES Modes	I/F	LCD/ Keypad
CN4010	ARM Cortex A9	A4010B	DC (Plug Pack)	1G Ethernet	CTR	RJ45	No
				1G Ethernet TIM			
CN4020	ARM Cortex A9	A4020B	DC (Plug Pack)	1G Ethernet		SFP	No
				1G Ethernet TIM			
CN6010	ARM Cortex A9	A6010B A6011B A6012B	AC/AC Dual DC/DC Dual AC/DC Dual	1G Ethernet		RJ45 SFP	Yes
				1G Ethernet TIM			
CN6110	ARM Cortex A9	A6110B A6111B A6112B	AC/AC Dual DC/DC Dual AC/DC Dual	1G Ethernet		RJ45 SFP+	Yes
				1G Ethernet TIM			
				10G Ethernet			
				10G Ethernet TIM			
CN6140	ARM Cortex A9	A6140B	AC/AC Dual	1Gx1 Ethernet Single Port 1Gx4 Ethernet Multi Port		SFP+	Yes

Model	CPU & ASIC	Hardware	Power	Protocol / FPGA Bitstream	AES Modes	I/F	LCD/ Keypad
		A6141B	DC/DC Dual	1Gx1 Ethernet TIM Single Port 1Gx4 Ethernet TIM Multi Port			
				10Gx1 Ethernet Single Port 10Gx2 Ethernet Multi Port			
		A6142B	AC/DC Dual	10Gx1 Ethernet TIM Single Port 10Gx4 Ethernet TIM Multi Port			
				10Gx4 Ethernet Multi Port			
CN9120	ARM Cortex A9	A9120B A9121B A9122B	AC/AC Dual DC/DC Dual AC/DC Dual	100G Ethernet		QSFP 28	Yes

2.2 Version Verification

4 CN4010:

```
CN4010>version

Software:
    Version      : 5.5.0
    Build Number : 1724475711
    Build Date   : 24-Aug-2024
    Build Time   : 05:01:51

Library
    Build ID: 31224

Serial Number:
    Management Module: 00D01F091090
CN4010>
```

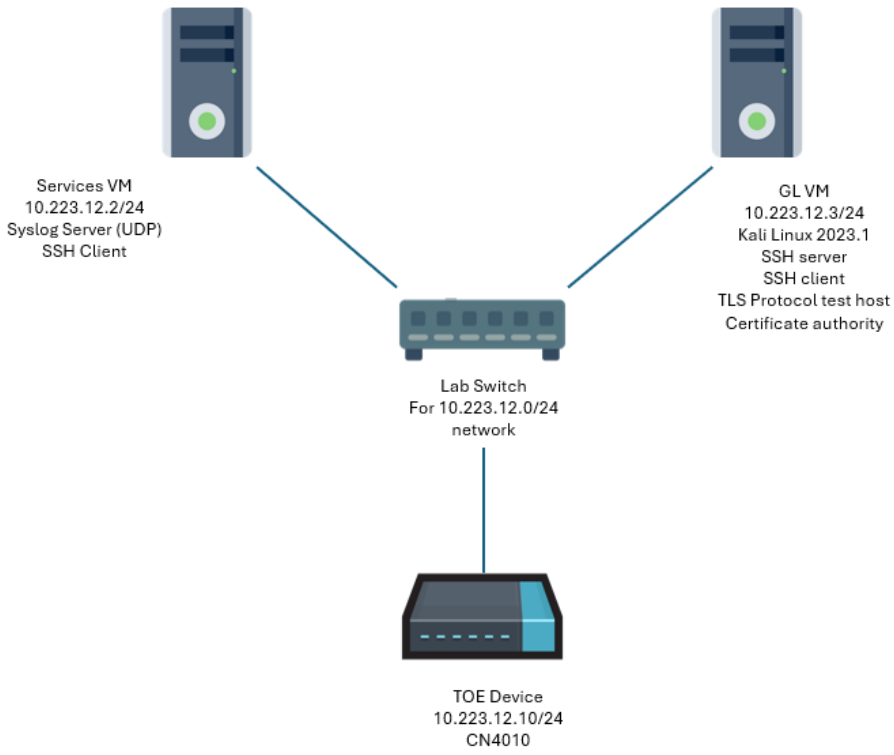
2.3 Non-TOE Components

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

a) **Audit Server.** The TOE can send audit events to a Syslog server.

2.4 Test Environment

6



7

8 Figure 1 shows a logical view of the test setup.

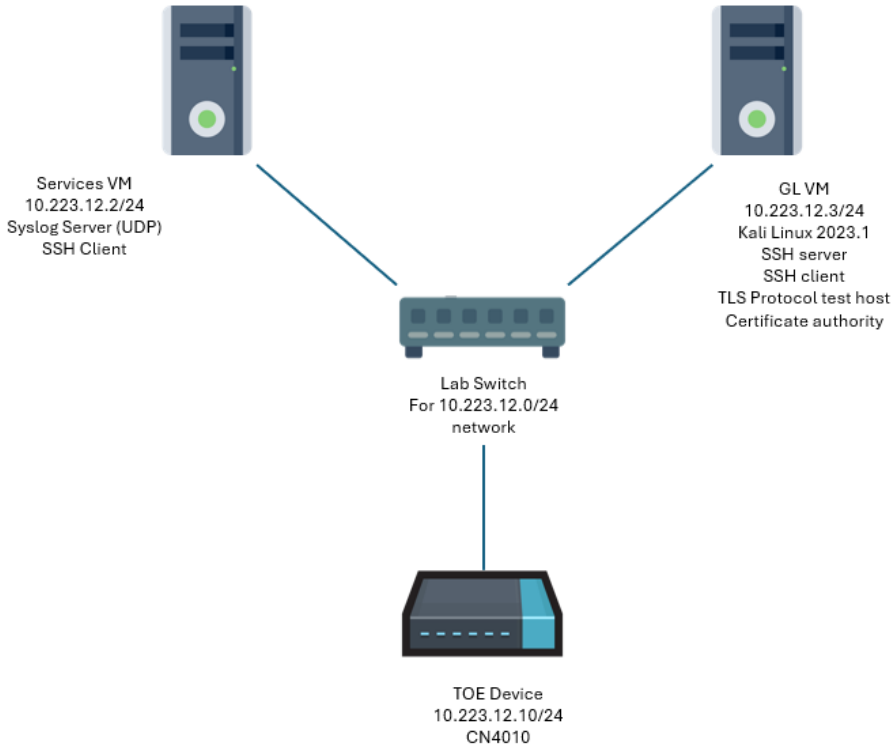


Figure 1 - Test setup

2.4.1 Logging

9 The Services VM was used as the logging server. Please see section 2.4.2 for additional details.

2.4.2 Systems

Table 5: Test Systems

Name / HW / SW	Description / Functions	Test Tools
CN4010 HW: ARM Cortex A9 SW: Senetas CN Series Encryptors 5.5.0 Build 31024, 31042 and 31224.	Fully tested TOE model TLS SSH * Most tests were conducted on build 31024 and 31042, while FPT_STM_EXT.1 and FPT_TUD_EXT.1 were tested on build 31224.	N/A

Name / HW / SW	Description / Functions	Test Tools
Services VM HW: Test Hypervisor SW: Debian 4.19.282-1	Logging Server (TLS) DNS Server	syslog-ng 3.19.1
Management Workstation HW: Test Hypervisor SW: 5.16.7-2kali1-amd64	SSH Client (SSH) Protocol Test Host (TLS/SSH) Certification Authority Perform Packet Captures	Greenlight 3.0.35 Python 3.11.4 OpenSSL 1.1.1m OpenSSH 8.8p1 Chrome 125.0.6422.114 Wireshark 4.0.8 tcpdump 4.99.1
Test Hypervisor HW: Dell PowerEdge R440 SW: ESXi, 7.0.3	Hypervisor for the Services VM and Management Workstation	None
Netgear Switch HW: ProSafe Plus GS105E	Physical disconnect packet captures	N/A
Packet Capture Laptop HW: Lenovo ThinkPad T15 SW: Windows 11	Physical disconnect packet captures	Wireshark 3.6.16

2.5 Test Platform Equivalency

2.5.1 Hardware Differences

10 Section 2.1 identifies the TOE models included in the evaluation.

11 All models of the TOE run the same firmware: 5.5.0 Build: 31224

12 The team used the [ND-SD] Network Device Equivalency Considerations as the basis for the following equivalency rationale:

Table 6: Equivalency Factors

Factor	Evaluator Guidance	Description
Platform/ Hardware Dependencies	If there are no identified platform/hardware dependencies, the evaluator shall consider testing on multiple hardware platforms to be equivalent. If there are specified differences between platforms/hardware, the evaluator must identify if the differences affect the cPP-specified security functionality or if they apply to non-cPP-specified functionality. If functionality specified in the cPP is dependent upon platform/hardware provided services, the product must be tested on each of the different platforms to be considered validated on that particular hardware combination. In these cases, the evaluator has the option of only re-testing the functionality dependent upon the platform/hardware provided functionality. If the differences only affect non-cPP-specified functionality, the variations may still be considered equivalent. For each difference the evaluator must provide an explanation of why the difference does or does not affect cPP-specified functionality.	Equivalent: There are no significant platform/hardware differences that would affect the operation of the TOE. The different models of the TOE use the same CPU and ASIC, ensuring consistent execution and performance across all models. The variations in power supply type do not impact the security functionality specified in the cPP, as the TOE's operation remains unaffected by these differences in power sources. Some models feature an LCD and Keypad, which are used for user interaction and do not influence the core security functions of the TOE. The LCD and Keypad are used for input/output operations that are peripheral to the main security functions, which are consistent across all models. There are differences in the network hardware and associated drivers. These differences are limited to network speed and physical layer differences. The higher level network operations remain the same on all models.
Differences in TOE Software Binaries	If the model binaries are identical, the model variations shall be considered equivalent. If there are differences between model software binaries, a determination must be made if the differences affect cPP-specified security functionality. If cPP-specified functionality is affected, the models are not considered equivalent and must be tested separately. The evaluator has the option of only retesting the functionality that was affected by the software differences. If the differences only affect non-PP specified functionality, the models may still be considered equivalent. For each difference the evaluator must provide an explanation of why the difference does or does not affect cPP specified functionality.	Equivalent: All models run the same binary.

Factor	Evaluator Guidance	Description
Differences in Libraries Used to Provide TOE Functionality	If there are no differences between the libraries used in various TOE models, the model variations shall be considered equivalent. If the separate libraries are used between model variations, a determination of whether the functionality provided by the library affects cPP-specified functionality must be made. If cPP-specified functionality is affected, the models are not considered equivalent and must be tested separately. The evaluator has the option of only retesting the functionality that was affected by the differences in the included libraries. If the different libraries only affect non-PP specified functionality, the models may still be considered equivalent. For each different library, the evaluator must provide an explanation of why the different libraries do or do not affect cPP specified functionality.	Equivalent: There are no differences between the libraries used in the different TOE models.
TOE Management Interface Differences	If there are no differences in the management interfaces between various TOE models, the model variations shall be considered equivalent. If the product provides separate interfaces based on the model variation, a determination must be made of whether cPP-specified functionality can be configured by the different interfaces. If the interface differences affect cPP-specified functionality, the variations are not considered equivalent and must be separately tested. The evaluator has the option of only retesting the functionality that can be configured by the different interfaces (and the configuration of said functionality). If the different management interfaces only affect non-PP specified functionality, the models may still be considered equivalent. For each management interface difference, the evaluator must provide an explanation of why the different management interfaces do or do not affect cPP specified functionality.	Equivalent: There are differences between the models with keypads. However, models with an LCD and Keypad provide additional methods for user interaction; however, these do not introduce separate management interfaces. The LCD and Keypad are supplementary components that facilitate input/output operations without impacting the core management functionality of the TOE. All cPP-specified functionalities are consistently configured and managed through the same interfaces across all models.
TOE Functional Differences	If the functionality provided by different TOE model variation is identical, the	Equivalent: There are no differences in functionality provided by different TOE models.

Factor	Evaluator Guidance	Description
	models variations shall be considered equivalent. If the functionality provided by different TOE model variations differ, a determination must be made if the functional differences affect cPP specified functionality. If cPP-specific functionality differs between models, the models are not considered equivalent and must be tested separately. In these cases, the evaluator has the option of only retesting the functionality that differs model-to-model. If the functional differences only affect non-cPP specified functionality, the model variations may still be considered equivalent. For each difference the evaluator must provide an explanation of why the difference does or does not affect cPP specified functionality.	

13 In summary, the evaluation team performed full testing on CN4010. All other models are considered equivalent to the tested models.

3 Evaluation Activities for Mandatory SFRs (NDcPPv3)

3.1 Security Audit (FAU)

3.1.1 FAU_GEN.1 Audit Data Generation

3.1.1.1 TSS

14 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 the logging of specific information to identify the relevant key: a) When importing a user public key, the action taken (importing) and the key reference. b) When deleting a user public key, the action taken (deleting) and the key reference. c) When generating a host key, the action taken (generating) and the key reference.

15 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

16 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 8, Log messages, provides an example of each auditable event required by FAU_GEN.1.

17 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 was 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

18 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
Reboot the TOE and verify the TOE logs the shutdown and startup of the audit function.
Findings
PASS

19 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.

20 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

21 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

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

23 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

24 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 6.1.3 states that the log files are transferred via SSH to the audit server in real time.

25 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 6.1.3 states that logs are stored locally, which means the TOE is a standalone component.

26 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 6.1.3 states that log files are transferred to the audit server in real time.

27 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.

28 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 6.1.3 describes the local storage methodology for logs, detailing their placement within rotating log files as outlined below:

a) Audit logs: Up to 4000 records are stored before they are rotated. Only one live log is kept.
b) Event logs: Up to 4000 records are stored before they are rotated. Only one live log is kept.

It also states that administrators may view audit records but are not provided with the capability to modify them.

29 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 6.1.3 describes the local logging method, mentioning the format of log files (audit and event), the persistence of audit logs.

30 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 6.1.3 states that the Administrator may view audit records, but no capability to modify the audit records is provided. Local audit logs are persistent.

31 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 6.1.3 states that the log rotation behavior is configurable, allowing for the option to either drop new records or overwrite the oldest records first. Log rotation takes place once the maximum number of records has been reached.

32 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

33 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 7.2, Logs, specifies that the encryptor can be configured to send log message to a remote syslog server, which establishes the trusted channel to the audit server. It describes the steps needed to secure the connection to the syslog server using SSH, including the requirement for ECDSA authentication of SSH key pairs.

34 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 7.2, Logs, states that the local log messages are sent to the remote audit server at the time of generation.

35 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
[AGD] Section 6.3 specifies that only a user with administrator privileges can acknowledge alarms or clear event and audit logs, which ensures protection against unauthorized modification or deletion.

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

EA Not Applicable	The storage size is not configurable.

37 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.

Findings
PASS
[AGD] Section 7.2 stated that when the local storage space becomes full, administrators have the option to configure the system to either overwrite the oldest records or to stop logging new records.

3.1.3.3 Tests

38 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).

39 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
Identify particular software (syslog-ng, version 3) of the audit server used during testing. Verification that audit data is not transferred in the clear is performed in conjunction with FTP_ITC.1 Test 3.
Findings
PASS

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 this data is stored locally and power cycle the TOE and ensure that following power on operations the log events generated are still maintained within the local audit storage.
Findings
PASS

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
Generate more than 4,000 entries in the log files to test the system. Verify that the TOE performs log rotation by deleting the oldest log file. Also, ensure that the TOE drops new audit data when the log wrapping function is disabled.
Findings
PASS

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.

High-Level Test Description
When the audit server is not available, the user with administrative privileges has access to locally stored audit records.
Findings
PASS

3.2 Cryptographic Support (FCS)

3.2.1 NIAP Policy 5

40 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	Algorithm in ST	Implementation name	Operational Environment	CAVP Alg.	CAVP Cert #
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;	CN Series Common Crypto Library	ARM Cortex A9	ECDSA KeyGen(FIPS 186-4) ECDSA KeyVer(FIPS186-4)	A3451
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”;	CN Series Common Crypto Library	ARM Cortex A9	KAS-ECC Sp800 56A/3 (NIST SP 800-56A Revision 3)	A3451

SFR	Algorithm in ST	Implementation name	Operational Environment	CAVP Alg.	CAVP Cert #
FCS_COP.1/ DataEncryption	encryption/decryption in accordance with a specified cryptographic algorithm AES used in [CTR] mode and cryptographic key sizes [128 bits, 256 bits]	CN Series Common Crypto Library	ARM Cortex A9	AES-CTR	A3451
FCS_COP.1/ SigGen	cryptographic signature services (generation and verification): RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits or greater]	CN Series Common Crypto Library	ARM Cortex A9	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	A3451
FCS_COP.1/ SigGen	cryptographic signature services (generation and verification): Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits or greater]	CN Series Common Crypto Library	ARM Cortex A9	ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)	A3451
FCS_COP.1/ Hash	cryptographic hashing services in accordance with a specified cryptographic algorithm [SHA-256, SHA-384, SHA-512]	CN Series Common Crypto Library	ARM Cortex A9	SHA-256 SHA2-384 SHA2-512	A3451
FCS_COP.1/ KeyedHash	keyed-hash message authentication in accordance with a specified cryptographic algorithm [HMAC-SHA-256, HMAC-SHA-512]	CN Series Common Crypto Library	ARM Cortex A9	HMAC-SHA2-256 HMAC-SHA2-512	A3451

SFR	Algorithm in ST	Implementation name	Operational Environment	CAVP Alg.	CAVP Cert #
FCS_RBG_EXT.1	deterministic random bit generation services using [selection: Hash_DRBG [SHA-256]	CN Series Common Crypto Library	ARM Cortex A9	Hash DRBG	A3451

3.2.2 FCS_CKM.1 Cryptographic Key Generation

3.2.2.1 TSS

41 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 states that the TOE supports key generation for the asymmetric schemes ECC P-256, ECC P-384, and ECC P-521. These schemes are utilized in SSH authentication and key exchange.

3.2.2.2 Guidance Documentation

42 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 7.1, CLI Access, states that the SSH server key pair is generated at boot time and remains persistent until an erase is performed. It also explains how to generate the SSH public key, including the restrictions (i.e., ECDSA) and notes that the TOE only allows NIST-approved curves.

3.2.2.3 Tests

43 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 RSA Schemes

44 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.

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

a) Random Primes:

- Provable primes
- Probable primes

b) Primes with Conditions:

- Primes p1, p2, q1, q2, p and q shall all be provable primes
- Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes
- Primes p1, p2, q1, q2, p and q shall all be probable primes

46 To test the key generation method for the Random Provable primes method and for all the 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. For each key length supported, the evaluator shall have the TSF generate 25 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.

Test Not Applicable	Key generation for FIPS PUB 186-4 RSA Scheme is not selected in FCS_CKM.1

Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

47 For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation.

FIPS 186-4 Public Key Verification (PKV) Test

48 For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Note	[ST] Table 4 specifies the CAVP certificate demonstrating the TOE correctly implements ECDSA key generation.

Key Generation for Finite-Field Cryptography (FFC)

49 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.

50 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

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

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

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

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

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

54 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.

55 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

56 for each FFC parameter set and key pair.

Test Not Applicable	The [ST] does not select FFC Schemes that meet FIPS PUB 186-4.

FFC Schemes using "safe-prime"

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

Test Not Applicable	The [ST] does not select FFC Schemes using 'safe-prime' groups that meet NIST Special Publication 800-56A Revision 3.

3.2.3 FCS_CKM.2 Cryptographic Key Establishment

3.2.3.1 TSS

58 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.

Findings
PASS
[ST] Section 6.2.2 states that ECC schemes are utilized in SSH key exchange, where the TOE functions as both the sender and receiver.

59 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 6.2.2 identifies the scheme being used by each service.

Scheme	SFR	Service
ECC	FCS_SSHS_EXT.1	Administration / Syslog

3.2.3.2 Guidance Documentation

60 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 7.1, CLI Access, provides instructions on how to generate an SSH key pair and configure remote CLI access via SSH. This section ensures secure key-based authentication for remote administration, which aligns with the TOE's key establishment process for secure communication

3.2.3.3 Tests

Key Establishment Schemes

61 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

62 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

63 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.

64 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.

65 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.

66 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.

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

Validity Test

68 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.

69 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).

70 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 4 specifies the CAVP certificate demonstrating the TOE correctly implements ECC SP 800-56A Revision 3 key agreement/establishment schemes.

RSA-based key establishment schemes

71 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 Schemes that uses RSAES-PKCS1-v1_5.

FFC Schemes using "safe-prime" groups

72 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 select FFC Schemes using 'safe-prime' groups that meet NIST Special Publication 800-56A Revision 3.

3.2.4 FCS_CKM.4 Cryptographic Key Destruction

3.2.4.1 TSS

73 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 6.2.3 provides information on how keys and CSPs are zeroized. Additionally, Table 17 in section 6.5.1 lists all relevant keys, their storage means, and how they are zeroized. The types of keys and CSPs are consistent with the claims made in section 5 of the [ST].

Key	Algorithm	Storage	Zeroization
SSH Private Keys	ECDSA	Flash – plaintext	SSH Private Keys are stored in plaintext in volatile storage are overwritten with zeros when destroyed.
SSH Public Keys	ECDSA	Flash-plaintext	SSH public keys are cleared from storage by Security Administrator initiated functions. Keys stored in plaintext in volatile storage are overwritten with zeros when destroyed.
SSH Session Keys	AES / ECDH	RAM – plaintext	OpenSSL ensures that keys (including re-keyed keys) are overwritten with zeroes when no longer required.
System Master Key (KEK)	AES / CFB	Flash – plaintext	The SMK is overwritten with zeroes when the erase button has been pressed.

74 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 6.2.3 states that key destruction of keys in non-volatile storage is initiated by the Security Administrator via the CLI/SSH administrative interfaces.

75 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 6.2.3 describes the zeroisation process, including the destruction of plaintext keys held in volatile storage and keys in non-volatile storage.

76 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 6.2.3 and the table provided in Section 6.5.1 indicates that all private key stored in non-volatile storage are stored in plaintext.

77 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.

EA Not Applicable	The [ST] TSS does not identify a configuration or circumstance that may not conform to the key destruction requirement.

78 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.

EA Not Applicable	The use of “a value that does not contain any CSP” is not included in the ST.

3.2.4.2 Guidance Documentation

79 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.

80 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).

EA Not Applicable	The [ST] does not identify a configuration or circumstance that may not conform to the key destruction.

[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

81 None

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

3.2.5.1 TSS

82 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 6.2.4 states that the TOE provides symmetric encryption and decryption capabilities using 128 and 256 bit AES in CTR mode. AES is implemented in SSH.

3.2.5.2 Guidance Documentation

83 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 7.7, Allowed algorithms, provides a list of each algorithm that the TOE supports, including the associated restrictions. [AGD] Section 7.1, CLI Access, explains how to generate the SSH public key, including any applicable restrictions.

3.2.5.3 Tests

AES-CBC Known Answer Tests

84 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.

85 **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.

86 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.

87 **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.

88 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.

89 **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].

90 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.

91 **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].

92 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

93 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.

94 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

95 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]
```

96 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.

97 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

98 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

a) **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.

b) **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.

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

99 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.

100 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.

101 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

102 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):

103 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.

104 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.

105 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.

106 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].

107 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

108 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

109 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]

110 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.

111 There is no need to test the decryption engine.

Note	[ST] Table 4 specifies the CAVP certificate demonstrating the TOE correctly implements AES.

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

3.2.6.1 TSS

112 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 6.2.5 specifies the cryptographic algorithm and key size supported by the TOE for signature services. Specifically, it supports: a) RSA Signature Algorithm with key size of 2048 bits, b) ECDSA Signature Algorithm with key sizes of 256, 384 and 521 bits

3.2.6.2 Guidance Documentation

113 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 7.7, Allowed algorithms, provides a list of each algorithm that the TOE supports, including the associated restrictions. [AGD] Section 7.1, CLI Access, explains how to generate the SSH public key, including any applicable restrictions.

3.2.6.3 Tests

ECDSA Algorithm Tests

ECDSA FIPS 186-4 Signature Generation Test

114 For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation.

ECDSA FIPS 186-4 Signature Verification Test

115 For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.

RSA Signature Algorithm Tests

Signature Generation Test

116 The evaluator generates or obtains 10 messages for each modulus size/SHA combination supported by the TOE. The TOE generates and returns the corresponding signatures.

117 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

118 For each modulus size/hash algorithm selected, the evaluator shall 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 using a trusted reference implementation of the signature generation algorithm. Some of the public keys, *e*, messages, or signatures are altered so that signature verification should fail. For both the set of original messages and the set of altered messages: the modulus, hash algorithm, public key *e* values, messages, and signatures are forwarded to the TOE, which then attempts to verify the signatures and returns the verification results.

119 The evaluator shall verify that the TOE confirms correct signatures on the original messages and detects the errors introduced in the altered messages.

Note	[ST] Table 4 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

120 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 6.2.6 states that SHA is implemented in various parts of the TSF, such as a) SSH; b) Digital signature verification as part of trusted update validation; and c) Hashing of passwords in non-volatile storage.

3.2.7.2 Guidance Documentation

121 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 7.7, Allowed algorithms, provides a list of each algorithm that the TOE supports, including the associated restrictions.

3.2.7.3 Tests

122 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.

123 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

124 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

125 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

126 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

127 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

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 4 specifies the CAVP certificate demonstrating the TOE correctly implements Hashing.

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

3.2.8.1 TSS

128 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] Table 16 in section 6.2.7 specifies the values used by the HMAC function.

Algorithm	Block Size	Key Size	Digest Size
HMAC-SHA-256	512 bits	256 bits	256 bits
HMAC-SHA-512	1024 bits	512 bits	512 bits

3.2.8.2 Guidance Documentation

129 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 7.7, Allowed algorithms, provides a list of each algorithm that the TOE supports, including the associated restrictions.

3.2.8.3 Tests

130 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 4 specifies the CAVP certificate demonstrating the TOE correctly implements HMAC algorithms.

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

131 Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Appendix D of [NDcPPv3].

3.2.9.1 TSS

132 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 6.2.8 specifies that a HASH_DRBG is seeded from a SP 800-90B compliant hardware-based TRNG, with the TRNG providing conditioned entropy that is used to seed the DRBG with 256 bits of full entropy. Regarding the minimum entropy supplied, the statement does not explicitly mention it. However, since it states that the seed contains 256 bits of "full entropy," it implies that each bit in the seed contributes the maximum possible amount of entropy, resulting in a total of 256 bits of entropy in the seed. Thus, the minimum entropy supplied by the TRNG is 256 bits.

3.2.9.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 5, Secure startup of the TOE, states that the RNG is not user configurable. The hardware based true RNG will start up automatically at boot time and will run continuously.

3.2.9.3 Tests

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

135 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).

136 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.

137 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 4 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

138 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 6.3.2 describes the logon process for both remote and local authentication mechanisms supported by the TOE. Locally, the TOE prompts the user for a password credential. Remotely, the TOE is administered via SSH with public keys. Access to TOE administrative functionality is granted only after the administrative user presents the correct authentication credentials.

139 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 6.3.2 describes that the process for local access involves the user being prompted to provide a username initially. For remote access, the user is required to provide both the username and public key credentials simultaneously. This setup implies that only authentication actions are permitted, with no additional actions allowed beyond the authentication process.

140 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.

141 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

142 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] Section 5 describes how to change the default administrator credentials.
[AGD] Section 7.1, CLI Access, describes the procedure for accessing the TOE locally, generating an SSH key pair, and installing the public key on the TOE for remote CLI access.

3.3.1.3 Tests

143 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 method of administration (serial and SSH) verify that attempting to log into the TOE with correct I&A credentials (username and password or SSH public key) allows access and invalid I&A credentials (invalid username, invalid password, or invalid SSH public key) denies access.
Findings
PASS

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.

Note	Configuration of services available to a remote entity prior to authentication is tested in FTA_TAB.1 Test 1.

Note	FIA_UIA_EXT.1 Test 1 Steps # 7 and 9 show that authentication is required for SSH. The SSH protocol enforces authentication flow, so no services other than the warning banner can be offered prior to authentication, so the Remote CLI is not tested in this test.

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
In local console, examine and show that the device does not have any services configured prior to I&A other than a TOE banner by entering 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

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

144 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

145 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

146 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

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

EA Not Applicable	The TOE is not distributed.

3.4.2.2 Guidance Documentation

148 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 7.8, Firmware Updates, explains how to upgrade the firmware using a USB device.

149 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

150 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
Attempt to initiate an update without logging in (the TOE does not support non administrator accounts) and show that the attempt is not permitted.
Findings
PASS

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

151 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 6.4.4 states that users are required to login before being provided with access to any administrative functions. Access to TSF data and functions is restricted to Security Administrator as described by FMT_SMR.2.

152 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	X.509v3 certificates are not used in management interfaces.

3.4.3.2 Guidance Documentation

153 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
[AGD] Section 6, User Roles and Privileges, describes the users of the TSF-data manipulating functions. It indicates that the Administrator has the highest privilege level and is authorized to access all module services. Additionally, the Supervisor, Operator, and Upgrader roles are available, but their access is limited to configuring and managing the TOE.

154 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	X.509v3 certificates are not used in management interfaces.

3.4.3.3 Tests

155 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.

Note	No separate testing for FMT_MTD.1/CoreData is required.

3.4.4 FMT_SMF.1 Specification of Management Functions

156 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)

157 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
[ST] Section 6.4.6 lists all of the management functions and indicates they are available via the CLI which is accessible via the console and SSH.
The evaluator examined the [AGD] and the TOE. The evaluator confirmed that all management functions in FMT_SMF.1 are provided by the TOE.

158 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 6.3.2 defines the console as a 'direct serial connection.' Since serial connections are inherently local, no additional warnings are necessary.
[AGD] Section 7.1, CLI Access, describes that the Administrators can access the TOE via the local serial CLI. No additional warnings are necessary. Once the terminal is correctly connected to the local console port the user will be prompted for their user credentials.

159 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.

160 (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 6.1.3 explains that log files are transferred to the audit server in real-time via SSH. It also details the local storage methodology for logs, specifying their placement within rotating log files as outlined below: a) Audit logs: Up to 4000 records are stored before they are rotated. Only one live log is kept. b) Event logs: Up to 4000 records are stored before they are rotated. Only one live log is kept. [AGD] Section 7.2, Logs, explains the logging implementation and how log files are maintained on the TOE.

3.4.4.2 Guidance Documentation

161 See [ND-SD] section 2.4.4.1.

3.4.4.3 Tests

162 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.

Note	FMT_SMF.1 is exercised throughout the test plan. No separate testing for FMT_SMF.1 is required.

3.4.5 FMT_SMR.2 Restrictions on Security Roles

3.4.5.1 TSS

163 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 6.4.7 defines four roles for accessing the TSF and details each role involving the administration of the TOE.

3.4.5.2 Guidance Documentation

164 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 7.1, CLI Access, provides instructions for administering the TOE both locally and remotely, including generating an SSH key pair and importing the public key on the TOE.

3.4.5.3 Tests

165 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	There are no explicit test activities and therefore none are recorded here. Both the remote SSH CLI and local console CLI are tested throughout this test plan.

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

166 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 how the keys are protected and ensures they cannot be viewed through any interface designed for that purpose, by any enabled role.

3.5.1.2 Guidance Documentation

167 None

3.5.1.3 Tests

168 None

3.5.2 FPT_STM_EXT.1 Reliable Time Stamps

3.5.2.1 TSS

169 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 6.5.5 specifies that the TOE allows the Security Administrator to set the time manually, and it lists two specific functions that make use of time: audit record timestamps and session timeouts.

170 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 TOE does not obtain time from the underlying VS.

3.5.2.2 Guidance Documentation

171 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
[AGD] Section 5, Secure startup of the TOE, describes how to set the time. The TOE does not support use of an NTP server.

172 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 TOE does not obtain time from the underlying VS.

3.5.2.3 Tests

173 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/time in various combinations of forward/backward including all elements (day, month, year, hour, minute, second, etc.) and verify that time was changed accordingly.
Findings
PASS

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.

Test Not Applicable The TOE does not support NTP.

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.

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

174 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

175 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 6.5.3 provided outlines that start up tests conducted by the TOE, including firmware integrity tests and cryptographic known answer tests. These tests ensure the correct operation of cryptographic functionality and verify the authenticity of the TOE image. If any of these tests fail, the device enters a Secure Halt state, preventing further operation until the issue is resolved.

176 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

177 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 5, Secure startup of the TOE, identifies the self-tests, possible errors, and administrative actions to be taken in response to errors.

178 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

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

a) Verification of the integrity of the firmware and executable software of the TOE

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

180 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.

NIAP TD0836

181 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
Reboot the TOE and observe BIOS self-test in the serial CLI console, then verify all required self-tests were performed during startup.
Findings
PASS

182 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

183 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 6.5.4 states that the current firmware version can be queried through any administrative interface; however, the TOE does not support delayed activation.

184 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 6.5.4 describes the software update mechanisms, including digital signature verification, and the actions that occur in both successful and unsuccessful signature verification scenarios.

185 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.

186 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

187 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 2, Acceptance Procedure, describes the “version” command to show the currently active version. The TOE does not support delayed activation.

188 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 7.8, Firmware Updates, provides instructions on how to perform firmware updates and describes procedures for both successful and unsuccessful verification.

189 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.

190 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.

191 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	Certificate-based mechanism for software update is not claimed.

3.5.4.3 Tests

192 The evaluator shall perform the following tests:

a) 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
For each method of update verification; show the current version of the TOE, install a legitimate update of the TOE, and verify version is consistent with the newly installed version.
Findings
PASS

b) 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
Attempt to install a modified update with a valid signature, an update without a signature, and an update with an invalid signature. Verify each update attempt fails.
Findings
PASS

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

Note	The TOE only supports manual updates. The test cases above are not applicable to automatic checking of updates, since there are no images to install during an automatic check.

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

Test Not Applicable	The TOE is not a distributed TOE.

3.6 TOE Access (FTA)

3.6.1 FTA_SSL.3 TSF-Initiated Termination

3.6.1.1 TSS

195 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 6.6.2 explicitly mentions the capability for the Security Administrator to configure the TOE to terminate inactive remote interactive sessions after a specified period of time.

3.6.1.2 Guidance Documentation

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

Findings
PASS
[ST] Section 7.3, Role allocation, describes configuring remote administrative session termination and the associated inactivity period.

3.6.1.3 Tests

197 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 several inactivity timeout values. Verify the TOE terminates remote SSH sessions when the inactivity period has elapsed.
Findings
PASS

3.6.2 FTA_SSL.4 User-Initiated Termination

3.6.2.1 TSS

198 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 6.6.3 explicitly states how both the remote administrative session and, if applicable, the local administrative sessions are terminated.

3.6.2.2 Guidance Documentation

199 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 7.1, CLI Access, indicates administrators can terminate their own sessions by using the ‘logout’ command.

3.6.2.3 Tests

200 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 serial console and immediately log out. Verify that the session has been terminated.
Findings
PASS

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 CLI interface and immediately log out. Verify the session has been terminated.
Findings
PASS

3.6.3 FTA_TAB.1 Default TOE Access Banners

3.6.3.1 TSS

201 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 6.6.4 states that the TOE displays an administrator-configurable message to users prior to login at the CLI and SSH CLI. Additionally, Section 7.3 specifies that an advisory notice and a consent warning message displayed according to FTA_TAB.1.

3.6.3.2 Guidance Documentation

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

Findings
PASS
[ST] Section 7.1, CLI Access, describe how to configure the banner message.

3.6.3.3 Tests

203 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
Change the banner to any string. Prior to I&A of both local console and SSH, verify that the banner was modified and is presented.
Findings
PASS

3.7 Trusted path/channels (FTP)

3.7.1 FTP_ITC.1 Inter-TSF trusted channel

3.7.1.1 TSS

204 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 states that the trusted channel is initiated by the external IT entity and the TOE acts as the server.

3.7.1.2 Guidance Documentation

205 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 7.2, Logs, describes how to configure a remote syslog server using an SSH CLI tunnel, as well as the recovery instructions for when the connection is broken.

3.7.1.3 Tests

206 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.

207 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 trusted channel to the remote audit server, which is set up as per the evaluated configuration. It is constantly tested throughout the evaluation.

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.

High-Level Test Description
Examine a packet capture performed in a testing involving data, such as username, being transferred to audit server and verify that they are not sent in plaintext.
Findings
PASS

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.

208 Further assurance activities are associated with the specific protocols.

209 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.

210 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

211 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 6.7.2 specifies SSH CLI as one of the trusted paths for remote administration, which aligns with the requirement for remote TOE administration methods. Additionally, it mentions FCS_SSH_EXT.1 and FCS_SSHS_EXT.1, which refer to specific security requirements related to SSH.

3.7.2.2 Guidance Documentation

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

Findings

PASS

[ST] Section 7.1, CLI Access, provides instructions for establishing remote administrative sessions through SSH.

3.7.2.3 Tests

213 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 trusted paths is the SSH Remote CLI, which is set up as per the evaluated configuration. It is constantly tested throughout the evaluation. SSH is tested in FCS_SSHS_EXT.1.

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

High-Level Test Description
Capture traffic while logging into the TOE over the trusted path. Verify the username and password are not sent in plaintext.
Findings
PASS

214 Further assurance activities are associated with the specific protocols.

215 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 (NDcPPv3)

216 No optional requirements have been selected.

5 Evaluation Activities for Selection-Based Requirements (NDcPPv3)

5.1 Identification and Authentication (FIA)

5.1.1 FIA_UAU.7 Protected Authentication Feedback

5.1.1.1 TSS

217 None

5.1.1.2 Guidance Documentation

218 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
[AGD] does not include any configuration steps to ensure authentication data is not revealed at the local console. While performing testing, the evaluator confirmed that no configuration is necessary.

5.1.1.3 Tests

219 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.

High-Level Test Description
Login to the local console and verify at most obscured feedback of the password entry is provided.
Findings
PASS

5.1.2 FIA_PMG_EXT.1 Password Management

5.1.2.1 TSS

220 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 lists the supported special characters for composing administrator passwords, and the minimum password length is configurable by the Administrator, ranging between 8 to 29 characters.

5.1.2.2 Guidance Documentation

221 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 7.3, Role allocation, identifies the characters that may be used in passwords, provides guidance on composing strong passwords, and specifies the minimum valid password length. Additionally, it gives instructions on setting the minimum password length.

5.1.2.3 Tests

222 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
Configure the TOE to enforce a password minimum length of 8, then set a password that meets the minimum password length exactly. Verify the password is accepted and can be used to login. Configure a password using all claimed characters. Verify the password is accepted and can be used to login.
Findings
PASS

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
The evaluator attempts to compose passwords that do not meet the requirements and verifies that the TOE does not support the invalid password combinations.
Findings
PASS

5.2 Protection of the TSF (FPT)

5.2.1 FPT_APW_EXT.1 Protection of Administrator Passwords

5.2.1.1 TSS

223 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 6.5.2 specifies that passwords are protected according to Table 18, which includes details on password generation, algorithm, and storage method. Specifically, it mentions that locally stored administrator passwords are user-generated and stored using Flash with SHA-512 hash encryption and AES-CFB 256-bit key encryption. Additionally, the ST states that plaintext passwords cannot be viewed through an interface designed specifically for that purpose.

5.3 Security management (FMT)

5.3.1 FMT_MOF.1/Services Management of Security Functions Behaviour

5.3.1.1 TSS

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

225 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 6.4.3 indicates the Security Administrator is able to start and stop the SSH service.

5.3.1.2 Guidance Documentation

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

227 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 7.1, CLI Access, describes how the SSH service can be enabled or disabled using the 'sshcli -e' command.

5.3.1.3 Tests

228 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.

Note	This activity is performed in conjunction with the testing of FMT_MOF.1/ManualUpdate Test 1.

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
Attempt to disable and enable remote-syslog with prior authentication as a Security Administrator and show that this can be done.
Findings
PASS

5.3.2 FMT_MOF.1/Functions Management of Security Functions Behaviour

5.3.2.1 TSS

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

230 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 states that the TOE restricts the ability to modify audit functionality when Local Audit Storage is full to Security Administrators.

5.3.2.2 Guidance Documentation

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

232 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 7.2, Logs, describes configuring a remote syslog server using the SSH CLI channel. Authentication for Syslog over SSH is restricted to ECDSA.

5.3.2.3 Tests

233 **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**

234 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.

Note	The ST does not claim this functionality.

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.

Note	The ST does not claim this functionality.

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

236 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.

Note	The ST does not claim this functionality.

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	The ST does not claim this functionality.

237 **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**

238 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.

High-Level Test Description
Attempt to modify the Syslog configuration without prior authentication as a Security Administrator and show that this cannot be done.
Findings
PASS

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.

High-Level Test Description
Fill the Local Audit Storage space and attempt to modify its behavior. Verify if the attempt is successful.
Findings
PASS

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

240 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.

Note	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.

Note	The ST does not claim this functionality.

5.3.3 FMT_MTD.1/CryptoKeys Management of TSF Data

5.3.3.1 TSS

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

242 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 6.4.5 states that the TOE restricts the ability to manage the cryptographic keys to Security Administrators and the Administrator is able to manage the import and deletion of the trusted public keys database for the purpose of remote SSH authentication.

5.3.3.2 Guidance Documentation

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

244 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 7.1 describes managing the SSH public keys.

5.3.3.3 Tests

245 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
Attempt to delete the existing SSH keys from the TOE without prior authentication and show the attempt is not successful.
Findings
PASS

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
As the privileged user, attempt to delete an SSH public key and show it does succeed.
Findings
PASS

5.4 TOE Access (FTA)

5.4.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

5.4.1.1 TSS

246 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 specifies that the Security Administrator may configure the TOE to terminate an inactive local interactive session after a specified period of time. The idle timeout for the local CLI can be configured between 3 to 60 minutes.

5.4.1.2 Guidance Documentation

247 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 7.3, Role allocation, describes configuring local administrative session termination and the associated inactivity period.

5.4.1.3 Tests

248 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 several inactivity timeout values. Verify the TOE terminates local console sessions when the inactivity period has elapsed.
Findings
PASS

6 Evaluation Activities for Mandatory SFRs (PKG_SSH)

6.1 Cryptographic Support (FCS)

6.1.1 FCS_SSH_EXT.1 SSH Protocol

6.1.1.1 FCS_SSH_EXT.1.1

6.1.1.1.1 TSS

249 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
[ST] Section 6.2.9 clearly outlines the selections made regarding the implementation of SSH for remote administration, including encryption algorithms, key exchanges, data integrity mechanisms, packet size limitations, and user authentication methods. These selections are consistent with the SFR requirement.

6.1.1.1.2 Guidance

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

EA Not Applicable	There are no guidance evaluation activities for this component.

6.1.1.1.3 Tests

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

Note	No tests have been defined.

6.1.1.2 FCS_SSH_EXT.1.2

6.1.1.2.1 TSS

252 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 6.2.9 clearly outlines the authentication methods supported by the TOE, which include public key authentication as described in the TSS. Additionally, the ST specifies the SSH key exchange methods and encryption algorithms used by the TOE, which align with the requirements stated in the SFR.

6.1.1.2.2 Guidance

253 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 7.1, CLI Access, states that public key authentication is the only authentication method supported for SSH.

6.1.1.2.3 Tests

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

255 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.

256 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.

High-Level Test Description
Using an SSH client, attempt to log into the TOE, ensuring that only the configured authentication methods for the TOE are offered by the server. Verify that the connections are successful only when using the supported authentication method.
Findings
PASS

257 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.

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

Note	The TOE is not acting as SSH Client.

259 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.

Note	The TOE is not acting as SSH Client.

260 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.

Note	The TOE is not acting as SSH Client.

6.1.1.3 FCS_SSH_EXT.1.3

6.1.1.3.1 TSS

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

Findings
PASS
[ST] Section 6.2.9 states that the TOE examines the size of each received SSH packet, and if the packet exceeds 256KB, it is automatically dropped.

6.1.1.3.2 Tests

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

High-Level Test Description
Using an SSH connection, log into the TOE with a valid public key. Ensure that the maximum allowed packet size is transmitted and verify that the connection is not terminated.
Findings
PASS

NIAP TD0732

263 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
Using a custom SSH client, log into the TOE using a valid public key but ensure that a large packet is transmitted and verify the connection is terminated.
Findings
PASS

6.1.1.4 FCS_SSH_EXT.1.4

6.1.1.4.1 TSS

264 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 6.2.9 describes the implementation of SSH in compliance with specific RFCs and lists the encryption algorithms, key exchanges, and authentication methods supported by the TOE. The ST mentions that the TOE implements SSH in compliance with RFCs 4251, 4252, 4253, 4254, 4344, 5656, and 6668. It also specifies the encryption algorithms, key exchanges, and authentication methods supported by the TOE.

6.1.1.4.2 Guidance

265 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 7.1, CLI Access, states that authentication for remote CLI access via SSH is restricted to ECDSA. It also provides a list of NIST-approved curves.

6.1.1.4.3 Tests

266 The evaluator shall perform the following tests.

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

268 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.

269 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
Establish an SSH connection to the TOE from a client and use a packet capture application to show that the communication is successful and the TOE encryption algorithms include only the algorithms claimed in the ST. Ensure that there are no additional algorithms claimed by the implementation that differ from the ST requirements.
Findings
PASS

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

High-Level Test Description
For the connection established in Test 1, the user logs out of the TOE and observe that the TOE terminates the connection.
Findings
PASS

271 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
Using an SSH client, forcibly attempt to negotiate an SSH host key using an unsupported host key algorithm and show it is unsuccessful.
Findings
PASS

6.1.1.5 FCS_SSH_EXT.1.5

6.1.1.5.1 TSS

272 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 6.2.9 specifies the supported hashing algorithms, hmac-sha2-256 and hmac-sha2-512, which align with the SFR requirement.

6.1.1.5.2 Guidance

273 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 7.1, CLI Access, explains how to generate the SSH public key and outlines the algorithms allowed by the TOE.

6.1.1.5.3 Tests

274 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
Examine the packet capture from FCS_SSH_EXT.1.4 Test 1 and verify the TOE utilizes HMAC-SHA2-256 and HMAC-SHA2-512, as claimed in the ST.
Findings
PASS

275 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
Using an SSH client, forcibly negotiate a hashing algorithm which is not supported by the TOE and show that it results in a failed connection.
Findings
PASS

6.1.1.6 FCS_SSH_EXT.1.6

6.1.1.6.1 TSS

276 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 6.2.9 specifies the supported shared secret establishment algorithms and confirms that those algorithms are identical to those listed for this component.

6.1.1.6.2 Guidance

277 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 7.7, Allowed algorithms, provides a list of each algorithm that the TOE supports, including the associated restrictions.

6.1.1.6.3 Tests

278 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
Examine the packet capture from FCS_SSH_EXT.1.4 Test 1 to verify that the appropriate mechanisms are advertised.
Findings
PASS

279 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
Using an SSH client, forcibly negotiate a key exchange method which is not supported by the TOE and show that it results in a failed connection.
Findings
PASS

6.1.1.7 FCS_SSH_EXT.1.7

6.1.1.7.1 TSS

280 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 6.2.9 states that the TOE uses the SSH KDF defined in RFC 5656 (Section 4), which implies that the KDFs supported are specified and are consistent with what is listed in the SFR.

6.1.1.8 FCS_SSH_EXT.1.8

6.1.1.8.1 TSS

281 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.

282 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.

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

Findings
PASS
[ST] Section 6.2.9 specifies that the TOE will rekey SSH connections after 1 hour or after an aggregate of 1 gigabyte of data has been exchanged, whichever occurs first.

6.1.1.8.2 Guidance

284 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.

Test Not Applicable	The connection rekey or termination limits are not configurable.

6.1.1.8.3 Tests

285 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.

286 Test 1: Establish an SSH connection. Wait until the identified connection rekey limit is met.

287 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, log into the TOE and keep the connection open for over 1 hour. Verify the TOE initiates a rekey of the SSH session when the time-based threshold is reached.
Findings
PASS

288 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
Connect to the TOE and transmit large amounts of data to the custom SSH client that the TOE rekeys before 1 GB in the aggregate has been transmitted. Ensure that the TOE is responsible for sending the rekey initiation.
Findings
PASS

289 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 large amounts of data over the channel. Ensure that the TOE rekeys before 1 GB in the aggregate has been transmitted. Ensure that the TOE is responsible for sending the rekey initiation.
Findings
PASS

7 Evaluation Activities for Optional SFRs (PKG_SSH)

7.1 Strictly Optional Requirements

This Package does not define any Strictly Optional requirements.

7.2 Objective Requirements

This Package does not define any Objective requirements.

7.3 Implementation-based Requirements

This Package does not define any Implementation-based requirements.

8 Evaluation Activities for Selection-based SFRs (PKG_SSH)

8.1 Cryptographic Support (FCS)

8.1.1 FCS_SSHS_EXT.1 SSH Protocol – Server

The inclusion of this selection-based component depends upon a selection in FCS_SSH_EXT.1.1.

8.1.1.1 FCS_SSHS_EXT.1

8.1.1.1.1 TSS

290 No activites.

8.1.1.1.2 Guidance

291 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 7.7, Allowed algorithms, provides a list of each algorithm that the TOE supports, including the associated restrictions.

8.1.1.1.3 Tests

NIAP TD0682

292 The evaluator shall perform the following tests:

293 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 SSH client, log into the TOE using the claimed public key algorithms with a valid key and show that the communication is successful.
Findings
PASS

294 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.

Note	This function is tested in FCS_SSHS_EXT.1 Test 1. The TOE only claims support for one authentication method.

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, forcibly negotiate an authentication mechanism which is not supported by the TOE and show that it results in a failed connection.
Findings
PASS

9 Evaluation Activities for Security Assurance Requirements

9.1 ASE: Security Target Evaluation

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

295 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.

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

296 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.

9.2 ADV: Development

9.2.1 Basic Functional Specification (ADV_FSP.1)

297 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.

298 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.

299 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.

300 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.

9.2.1.1 Evaluation Activity

301 *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.*

302 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.

303 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 [NDcPPv3]: “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.

9.2.1.2 Evaluation Activity

304 *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.

9.2.1.3 Evaluation Activity

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

306 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.

307 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.

308 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 [NDcPPv3]: “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.

9.3 AGD: Guidance Documents

309 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.

310 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)

9.3.1 Operational User Guidance (AGD_OPE.1)

311 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.

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

9.3.1.1 Evaluation Activity

313 *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
[AGD] Section 2 explains that the TOE FW and User Guides are available to users with a maintenance contract via the Senetas customer portal (https://support.senetas.com/). Users without a maintenance contract can access the FW and User Guides using a onetime temporal link on the Senetas SureDrop secure file sharing platform, provided by Senetas.

9.3.1.2 Evaluation Activity

314 *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 2.2.2, Figure 2. All TOE platforms claimed in [ST] are covered by the operational guidance. This is evidenced by the platform equivalency.

9.3.1.3 Evaluation Activity

315 *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] Section 7, Rules for secure usage of the TOE, provides instructions for configuring any cryptographic implementations associated with the evaluated configuration of the TOE. It also specifies that only NIST-approved algorithms are permitted.

9.3.1.4 Evaluation Activity

316 *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
[AGD] Section 7, Rules for secure usage of the TOE, clarifies the evaluated functionality. The evaluator confirmed [AGD] covers configuration of the in-scope functionality where additional configuration might be required.

9.3.1.5 Evaluation Activity

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

a) 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.

b) 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).

c) 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 9.3.1.3 for configuration of the cryptographic engine. [AGD] section 7.8 describes the update process. See section 9.3.1.4 for details as to what was covered by the EAs.

9.3.2 Preparative Procedures (AGD_PRE.1)

318 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.

319 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.

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

9.3.2.1 Evaluation Activity

321 *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).*

322 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
[AGD] is written in a style that an average IT administrator (with general security knowledge, but not a CC/Senetas expert) can understand the steps that need to be performed to configure the TOE.

9.3.2.2 Evaluation Activity

323 *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 [ST] section 2.2.2, Figure 2. All TOE platforms claimed in [ST] are covered by the operational guidance. This is evidenced by the platform equivalency.

9.3.2.3 Evaluation Activity

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

Findings
PASS
There is only one operational environment claimed in [ST] section 2.2.2, Figure 2. All TOE platforms claimed in [ST] are covered by the operational guidance. This is evidenced by the platform equivalency. While performing testing, the evaluator ensured the instructions are sufficient to successfully install the TOE in the operational environment.

9.3.2.4 Evaluation Activity

325 *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.*

326 *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
[AGD] Section 3 describes the setup of a secure operational environment and provides instructions on how to manage the TSF within this environment. The guidance documentation provides extensive information on managing the security of the TOE as an individual product.

9.3.2.5 Evaluation Activity

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

328 The preparative procedures must:

a) include instructions to provide a protected administrative capability; and

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

Findings
PASS
[AGD] Section 7.1 describes the protected administrative capability over SSH. [AGD] Section 5 includes the default TOE passwords and instructions on how to create a new administrator account to guard against unauthorized access.

9.4 ALC: Life-cycle Support

9.4.1 Labelling of the TOE (ALC_CMC.1)

329 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
While performing the ALC_CMC.1 CEM work units, the evaluator verified the TOE is labeled with a unique reference and the reference is consistent with the ST.

9.4.2 TOE CM Coverage (ALC_CMS.1)

330 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
While performing the ALC_CMS.1 CEM work units, the evaluator verified the configuration list contains the TOE and the evaluation evidence required by the SARs. Each configuration item was determined to contain a unique identifier.

9.4.3 Basic Flaw Remediation (ALC_FLR.1) (optional)

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

EA Not Applicable	The work unit is not selected in the CEM.

9.4.4 Flaw Reporting Procedures (ALC_FLR.2) (optional)

332 When evaluating the developer's flaw reporting procedures, the evaluator performs the work units as presented in the CEM.

EA Not Applicable	The work unit is not selected in the CEM.

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

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

EA Not Applicable	The work unit is not selected in the CEM.

9.5 ATE: Tests

9.5.1 Independent Testing – Conformance (ATE_IND.1)

334 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.

335 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.

336 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.

337 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.

9.6 Vulnerability Assessment

9.6.1 Vulnerability Survey (AVA_VAN.1)

338 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.

339 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.

340 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.

9.6.1.1 Evaluation Activity (Documentation)

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

342 *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.*

343 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).

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

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

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

c) 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.

9.6.1.2 Evaluation Activity

345 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:
- Senetas security advisories (https://support.senetas.com/)
- CVEs
  o NIST National Vulnerabilities Database (can be used to access CVE and US-CERT databases identified below): https://web.nvd.nist.gov/view/vuln/search
  o Common Vulnerabilities and Exposures: http://cve.mitre.org/cve/
  o Common Vulnerabilities and Exposures: https://www.cvedetails.com/vulnerability-search.php
- US-CERT: http://www.kb.cert.org/vuls/html/search
- Tenable Network Security http://nessus.org/plugins/index.php?view=search
- Tipping Point Zero Day Initiative http://www.zerodayinitiative.com/advisories
- Offensive Security Exploit Database: https://www.exploit-db.com/
- Rapid7 Vulnerability Database: https://www.rapid7.com/db/vulnerabilities

Type 1 Hypothesis searches were conducted on December 12, 2024 and included the following search terms:

- Senetas CN Series Encryptors 5.5.0
- Each TOE hardware model
- ARM Cortex A9
- OpenSSL
- OpenSSH
- Common Crypto Library
- Coreutils
- Bash
- Curl
- Net-snmp
- Microhttpd
- Ulfius
- Pamtacplus
- Keysecure
- Busybox
- ppp
- lcd4linux
- Liboqs
- Liboqse
- Lxc
- Serdisplib

Note: Additional proprietary search terms were also included.

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

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.

9.7 Evaluating Additional Components for a Distributed TOE

346 In the case of a distributed TOE the Security Target will identify an evaluated configuration that consists of a number of separate components chosen by the ST author, which collectively satisfy the requirements of the cPP. This evaluated configuration need not be the minimum set of components that could possibly meet the cPP (e.g. if the TOE is intended for large enterprise deployments then the evaluated configuration might include some redundancy in components in order to support expected connectivity and loads), but because this is the main configuration referred to in the ST and the evaluation, it is treated in this section as the minimum configuration of interest and is referred to here as the 'minimum configuration' as well as the 'evaluated configuration'.

347 In addition to the minimum configuration above, the ST may also identify (at the author's discretion, and subject to verification as described in this section) which TOE components can have instances added to an operational configuration without affecting the validity of the CC certification. The ST description may include constraints on how such components are added, including required and/or prohibited configurations of the components.

348 Extra instances of a TOE component must have the same hardware and software as the original component included in the evaluated configuration.

349 It is noted that undesirable configurations may be possible in the operational deployment of a TOE – such as allowing a TOE component to be managed from separate and potentially conflicting administration domains. However, the definition of 'undesirable' and of the risks involved in such cases will be specific to each operational environment and is therefore not treated as part of the evaluation. Correct and appropriate configuration of this sort remains a matter for expert network planning and design in the operational environment.

9.7.1 Evaluator Activities for Assessing the ST

9.7.1.1 TSS

350 The evaluator shall examine the TSS to confirm it identifies any extra instances of TOE components allowed in the ST and what effects will occur when extra instances of distributed TOE components are added. The information in the TSS shall allow the evaluator to understand how a system with one component behaves in comparison to a system with multiple components. The TSS also shall describe how the additional components maintain the SFRs to determine it is consistent with the role the component plays in the evaluated configuration and cannot be used in a way that the security functionality would be corrupted or bypassed. In general, any additional TOE-component shouldn't have a negative impact on other components that are already part of the TOE.

Note	The TOE is not a distributed TOE.

9.7.2 Evaluator Activities for Assessing the Guidance Documentation

9.7.2.1 Guidance Documentation

351 The evaluator shall examine the description of the extra instances of TOE components in the guidance documentation to confirm that they are consistent with those identified as allowed in the ST. This includes confirmation that the result of applying the guidance documentation to configure the extra component will leave the TOE in a state such that the claims for SFR support in each component are as described in the ST and therefore that all SFRs continue to be met when the extra components are present.

Note	The TOE is not a distributed TOE.

352 The evaluator shall examine the secure communications described for the extra components to confirm that they are the same as described for the components in the minimum configuration (additional connections between allowed extra components and the components in the minimum configuration are allowed of course).

Note	The TOE is not a distributed TOE.

9.7.3 Evaluator Activities for Testing the TOE

9.7.3.1 Tests

353 The evaluator shall test the TOE in the minimum configuration as defined in the ST (and the guidance documentation).

354 If the description of the use of extra components in the ST and guidance documentation identifies any difference in the SFRs allocated to a component, or the scope of the SFRs involved (e.g. if different selections apply to different instances of the component) then the evaluator shall test these additional SFR cases that were not included in the minimum configuration.

355 In addition, the evaluator shall test the following aspects for each extra component that is identified as allowed in the distributed TOE:

- Communications: the evaluator shall follow the guidance documentation to confirm, by testing, that any additional connections introduced with the extra component and not present in the minimum configuration are consistent with the requirements stated in the ST (e.g. with regard to protocols and ciphersuites used). An example of such an additional connection would be if a single instance of the component is present in the minimum configuration and adding a duplicate component then introduces an extra communication between the two instances. Another example might be if the use of the additional components necessitated the use of a connection to an external authentication server instead of using locally stored credentials.

Note	The TOE is not a distributed TOE.

- Audit: the evaluator shall confirm that the audit records from different instances of a component can be distinguished so that it is clear which instance generated the record.

Note	The TOE is not a distributed TOE.

- Management: if the extra component manages other components in the distributed TOE then the evaluator shall follow the guidance documentation to confirm that management via the extra component uses the same roles and role holders for administrators as for the component in the minimum configuration.

Note	The TOE is not a distributed TOE.