CAE

MPIC 3.0.66

Assurance Activity Report

Version 0.6

November 2022

Document prepared by

www.lightshipsec.com

Table of Contents

1 Introduction

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

1.1 Evaluation Identifiers

Table 1: Evaluation Identifiers

Scheme	NIAP
Evaluation Facility	Lightship Security USA 3600 O'Donnell St., Suite 2 Baltimore, MD 21224
Developer/Sponsor	CAE Inc. 8585, Ch. de la Cote-de-Liesse St-Laurent, QC H4T 1G6 Canada
TOE	MPIC 3.0.66
Security Target	MPIC 3.0.66 Security Target, Version 1.10, October 2022
Protection Profile	collaborative Protection Profile for Network Devices, v2.2E (NDcPP), 23-March-2020

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, v2.2 (NDcPP-SD)
Interpretations	**NDcPP v2.2e**

TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1) *N/A: The TOE does not use X509 certificates.*
TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4
TD0536: NIT Technical Decision for Update Verification Inconsistency
TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3 *N/A: The TOE does not implement a TLS Client communication channel.*
TD0538: NIT Technical Decision for Outdated link to allowed-with list
TD0546: NIT Technical Decision for DTLS - clarification of Application Note 63 *N/A: The TOE does not implement a DTLS communication channel.*
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test *N/A: The TOE does not implement a TLS Server communication channel.*
TD0556: NIT Technical Decision for RFC 5077 question *N/A: The TOE does not implement a TLS Server communication channel.*
TD0563: NiT Technical Decision for Clarification of audit date information
TD0564: NiT Technical Decision for Vulnerability Analysis Search Criteria
TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7 *N/A: The TOE does not implement a TLSS or DTLS communication channel.*
TD0570: NiT Technical Decision for Clarification about FIA_AFL.1
TD0571: NiT Technical Decision for Guidance on how to handle FIA_AFL.1
TD0572: NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers

	TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e
	TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3
	TD0591: NIT Technical Decision for Virtual TOEs and hypervisors *N/A: The TOE is not a virtual TOE.*
	TD0592: NIT Technical Decision for Local Storage of Audit Records
	TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server
	TD0632: NIT Technical Decision for Consistency with Time Data for vNDs *N/A: The TOE is not a virtual TOE.*
	TD0633: NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance *N/A: The TOE does not implement IPsec functionality.*
	TD0634: NIT Technical Decision for Clarification required for testing IPv6 *N/A: Neither TLSC nor DTLSC requirements are claimed.*
	TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters *N/A: The TOE does not implement a TLS Server communication channel.*
	TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH
	TD0638: NIT Technical Decision for Key Pair Generation for Authentication
	TD0639: NIT Technical Decision for Clarification for NTP MAC Keys
	TD0670: NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing *N/A: Neither TLSC nor DTLSC requirements are claimed.*
Tools	Please see associated Test Plan.

2 TOE Details

2.1 Overview

3 The Security Target (ST) defines the CAE MPIC Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

4 The CAE MPIC is a standalone physical Network Device, used to transmit data from the hardware panels to a software-based flight simulation, processed by one or more Daughter Boards (DB). The simulation data is processed by the DB's and then feedback is transmitted back to the hardware panels via the MPIC. It comes in a range of form factors MPIC, MPIC-PCMIP, MPIC-EMB. The different form factors can be installed in combination or independently to Network data. All form factors provide a basic set of security functions such as, a secure remote management path, identification and authentication services to trusted administrators, and secure auditing of administrator actions. The MPIC-PCMIP form factor differs as it has standard type slot for extensions compared to the custom interface on the MPIC. The MPIC-EMB differs as it is designed to be embedded and not mounted into systems.

2.2 TOE Models

Table 3: TOE models

Type	Model	CPU	Software	Differences
MPIC	MPIC	i.MX6 ARM Cortex-A9 (ARMv7-A) with CAAM	cae-mx6qmpic-3.0.66 MPICLinuxDistributionXR 3.0	Form Factor
	MPIC-PCMIP	i.MX6 ARM Cortex-A9 (ARMv7-A) with CAAM		
	MPIC-EMB	i.MX6 ARM Cortex-A9 (ARMv7-A) with CAAM		

2.3 Reference Documents

Table 4: List of Reference Documents

Ref	Document
[PP]	collaborative Protection Profile for Network Devices, Version 2.2e, 23-March-2020
[SD]	Supporting Document Mandatory Technical Document Evaluation Activities for Network Device cPP, December-2019, Version 2.2
[ST]	MPIC 3.0.66 Security Target, Version 1.10, October 2022
[AGD]	CAE Inc. MPIC v3.0.66 Common Criteria Guide, Version 1.1, October 2022

Ref	Document
[ADMIN]	Getting Started with MPIC Developer's Guide, TPD 20365 Rev 7, 20 Oct 2022
[DTR]	CAE MPIC v3.0.66 NDcPP 2.2E Test Plan, Version 0.6, November 2022 CAE MPIC v3.0.66 NDcPP 2.2E Test Results, Version 0.6, November 2022
[ETR]	CAE MPIC v3.0.66 Evaluation Technical Report, Version 0.6, November 2022

2.4 Summary of SFRs

Table 5: Summary of SFRs

Requirement	Title
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_NTP_EXT.1	NTP Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHC_EXT.1	SSH Client Protocol
FCS_SSHS_EXT.1	SSH Server Protocol
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FMT_MOF.1/Functions	Management of Security Functions Behaviour

Requirement	Title
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour
FMT_MOF.1/Services	Management of Security Functions Behaviour
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on Security Roles
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF Testing
FPT_TUD_EXT.1	Trusted Update
FPT_STM_EXT.1	Reliable Time Stamps
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_TAB.1	Default TOE Access Banners
FTP_ITC.1	Inter-TSF trusted channel
FTP_TRP.1/Admin	Trusted Path

3 Evaluation Activities for SFRs

3.1 Security Audit (FAU)

3.1.1 FAU_GEN.1 Audit data generation

3.1.1.1 TSS

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

Findings:	[ST] Section 6.1.1 - The following information is logged as a result of the Security Administrator generating/importing or deleting cryptographic keys: a) Generate SSH key-pair. Action and key reference. b) Import of SSH public key. Action and key reference. c) Deletion of SSH public key. Action and key reference

6 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.

Findings:	The TOE is not a distributed TOE.

3.1.1.2 Guidance Documentation

7 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:	[AGD] Section 4.2 contains a table listing the example audit events. The evaluator reviewed this table to confirm that all applicable auditable events are captured in this table.

8 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:	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 document in the form of [AGD]. However, AAs for guidance were performed combining all the Guidance documents listed in the **Table 4: List of Reference Documents** of this document.

3.1.1.3 Tests

9 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 I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the guidance documentation, and that the fields in each audit record have the proper entries.

High-Level Test Description
Ensure that the TOE displays an audit record for each of the auditable events defined for this requirement.
Finding: Pass. The evaluator confirmed that an audit record was generated by the TOE for each required auditable event.

10 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 a distributed TOE.

11 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

12 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

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

Note	These activities are performed in conjunction with the testing of FAU_GEN.1.1.

14 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 a distributed TOE.

3.1.3 FAU_STG_EXT.1 Protected audit event storage

3.1.3.1 TSS

15 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:	[ST] Section 6.1.3 - Log files are transferred via SSH tunnel (see FCS_SSHC_EXT.1) to the audit server.

16 The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access.

Findings:	[ST] Section 6.1.3 states “When local audit logs reach a maximum size of 8MB, logs are rotated out by removing the oldest log first and creating a new log file.” Additionally, audit logs are stores in /var/log/syslog and /var/audit/audit.log which preserve 4 weeks of backlogs and are rotated weekly. Section 6.1.3 also states that the audit logs cannot be modified and are only viewable by authorized administrators.

17 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:	The TOE is standalone and not a distributed TOE.

18 The evaluator shall examine the TSS to ensure that 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:	[ST] Section 6.1.3 - When local audit logs reach a maximum size of 8MB, logs are rotated out by removing the oldest log first and creating a new log file.

19 The evaluator shall examine the TSS to ensure that it details whether the transmission of audit information to an external IT entity can be done in real-time or periodically. In case the TOE does not perform transmission in real-time the evaluator needs to verify that the TSS provides details about what event stimulates the transmission to be made as well as the possible as well as acceptable frequency for the transfer of audit data.

Findings:	[ST] Section 6.1.3 states that log files are transferred in real time via SSH tunnel.

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

Findings:	The TOE is not a distributed TOE.

21 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.

Findings:	The TOE is not a distributed TOE.

3.1.3.2 Guidance Documentation

22 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:	[AGD] Section 3.6 states the configuration needed to establish and SSH connection to the audit server. This section references [ADMIN] which contains steps on configuring the trusted channel to the audit server via SSH.

23 The evaluator shall also examine the guidance documentation to determine that 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:	[AGD] Section 3.6 states that once remote logging is configured then the TOE sends logs to the remote host in real time.

24 The evaluator shall also ensure that the guidance documentation describes all possible configuration options for FAU_STG_EXT.1.3 and the resulting behaviour of

the TOE for each possible configuration. The description of possible configuration options and resulting behaviour shall correspond to those described in the TSS.

Findings:	[AGD] Section 3.6 “Audit logging” section says, “the oldest is overwritten when the backlog is full.” This is the only configuration supported.

3.1.3.3 Tests

25 Testing of the trusted channel mechanism for audit will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall perform the following additional tests 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
This test is performed in conjunction with FTP_ITC.1.
Finding: Pass. The evaluator confirmed that audit data is encrypted and sent to the audit server in conjunction with FTP_ITC.1.

b) Test 2: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. 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.1.3. Depending on the configuration this means that the evaluator has to check the content of the audit data when the audit data is just filled to the maximum and then verifies that

1) 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.1.3).

2) 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.1.3)

3) The TOE behaves as specified (for the option ‘other action’ in FAU_STG_EXT.1.3).

High-Level Test Description
Examine the last log entry and then perform a series of actions which will push the oldest record off the list.
Finding: Pass. The evaluator confirmed that when the local audit trail is full, the oldest audit record is removed to make space for the new record.

c) Test 3: If the TOE complies with FAU_STG_EXT.2/LocSpace the evaluator shall verify that the numbers provided by the TOE according to the selection for FAU_STG_EXT.2/LocSpace are correct when performing the tests for FAU_STG_EXT.1.3

Test Not Applicable	The TOE does not claim this functionality.

d) Test 4: For distributed TOEs, Test 1 defined above should be applicable to all TOE components that forward audit data to an external audit server. For the local storage according to FAU_STG_EXT.1.2 and FAU_STG_EXT.1.3 the Test 2 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 FAU_STG_EXT.2/LocSpace Test 3 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 a distributed TOE.

3.2 Cryptographic Support (FCS)

3.2.1 FCS_CKM.1 Cryptographic Key Generation

3.2.1.1 TSS

26 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:	[ST] Section 6.2.1 states that the TOE provides cryptographic signature generation services using: RSA key sizes: 2048, 3072, and 4096 bits ECC key sizes: 256, 384, and 521 FFC key sizes: 2048, 3072, and 4096 bits

3.2.1.2 Guidance Documentation

27 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:	[AGD] Section 3.3 covers how the RSA SSH host and public keys can be generated on the TOE. [ST] Section 6.2.1 states that ECC keys are only used for SSH key exchange; thus are not configurable.

3.2.1.3 Tests

28 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

29 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*.

30 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

31 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.

Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

32 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

33 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.

Key Generation for Finite-Field Cryptography (FFC)

34 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.

35 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

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

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

37 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.

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

39 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.

40 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

41 for each FFC parameter set and key pair.

[NIAP TD0580] ***FFC Schemes using "safe-prime" groups***

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

Findings:	The vendor uses the CAVP certificate A2558 for ECDSA and RSA key generation. These are described in [ST] Table 4.

3.2.2 FCS_CKM.2 Cryptographic Key Establishment

3.2.2.1 TSS

[NIAP TD0580]

43 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:	[ST] Section 6.2.2 states that the TOE supports the following key establishment schemes:

	Elliptic curves for SSH as sender and receiver. FFC using safe primes for SSH as sender and receiver using Diffie Hellman groups: 14, 16, and 18.

44 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:

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

45 The information provided in the example above 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.

Findings:	[ST] Section 6.2.2 provides table 13 which shows the following schemes used for each service: RSA: FCS_SSHS_EXT.1: Administration RSA: FCS_SSHC_EXT.1: Audit Server ECC: FCS_SSHS_EXT.1: Administration ECC: FCS_SSHC_EXT.1: Audit Server

3.2.2.2 Guidance Documentation

46 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:	[AGD] Section 3.3 states that FIPS mode is enabled by default and no further configuration is needed to achieve the evaluated cryptographic configuration.

3.2.2.3 Tests

[NIAP TD0580]

Key Establishment Schemes

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

SP800-56A Key Establishment Schemes

48 The evaluator shall verify a TOE's implementation of SP800-56A key agreement schemes using the following Function and Validity tests. 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

49 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.

50 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.

51 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.

52 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.

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

Validity Test

54 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.

55 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 will 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).

56 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.

RSA-based key establishment schemes

57 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.

FFC Schemes using "safe-prime" groups

58 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.

Findings:	The vendor uses the CAVP certificate A2558 for ECDSA and RSA key establishment. These are described in [ST] Table 4.

3.2.3 FCS_CKM.4 Cryptographic Key Destruction

3.2.3.1 TSS

59 The evaluator examines 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 confirms 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[1]). In particular, if a TOE claims not to store plaintext keys in non-volatile memory then the evaluator checks that this is consistent with the operation of the TOE.

Findings:	[ST] Section 6.5.1 table 15 contains all relevant keys which includes key purpose, algorithms used, storage type, and zeroization method.

60 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:	[ST] Section 6.2.3 and the table provided in section 6.5.1 in the [ST] lists how each key gets deleted and interfaces used to delete each key.

61 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

[1] Where keys are stored encrypted or wrapped under another key then this may need to be explained in order to allow the evaluator to confirm the consistency of the description of keys with the TOE functions.

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:	[ST] does not make the '*destruction of reference*' (for volatile memory) selection. Section 6.5.1 Table 15 contains how the SSH Private Keys and NTP Key (in non-volatile memory) are destroyed via invocation of a CLI command.

62 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:	[ST] Section 6.2.3 and the table provided in section 6.5.1 in the indicates that all private keys stored in non-volatile storage are stored in plaintext.

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

Findings:	[ST] Section 6.2.3 does not indicate any circumstances that may not conform to the key destruction requirements.

64 Where the ST specifies the use of "a value that does not contain any CSP" to overwrite keys, the evaluator examines 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.

Findings:	The [ST] does not claim this selection.

3.2.3.2 Guidance Documentation

65 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.

66 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[2] and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

Findings:	[AGD] Section 3.3 states the commands used to initiate key destruction on the TOE. Once these commands are executed the keys are destroyed. No situations where the key destruction may be delayed are identified.

[2] 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).

3.2.3.3 Tests

67 None

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

3.2.4.1 TSS

68 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:	[ST] Section 6.2.4 states that the TOE provides symmetric encryption and decryption capabilities using 128 and 256 bit AES in CTR mode.

3.2.4.2 Guidance Documentation

69 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:	[AGD] Section 3.3 states that FIPS mode is enabled by default and no further configuration is needed to achieve the evaluated cryptographic configuration.

3.2.4.3 Tests

AES-CBC Known Answer Tests

70 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.

71 **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.

72 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.

73 **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.

74 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.

75 **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].

76 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.

77 **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].

78 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

79 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.

80 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

81 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]
```

82 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.

83 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

84 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.

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

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

85 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.

86 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.

87 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

88 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 through FCS_SSH*_EXT.1.4. 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):

89 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.

90 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.

91 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.

92 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].

93 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

94 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

95 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]
```

96 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.

97 There is no need to test the decryption engine.

Findings:	The vendor uses the CAVP certificate A2558 for AES in CTR mode. This is described in [ST] Table 4.

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

3.2.5.1 TSS

98 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:	[ST] Section 6.2.5 states that the TOE provides cryptographic signature generation and verification services using: RSA with key sizes: 2048, 3072, and 4096 ECDSA with NIST curves: P-256, P-384, and P-521

3.2.5.2 Guidance Documentation

99 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:	[AGD] Section 3.3 states that FIPS mode is enabled by default and no further configuration is needed to achieve the evaluated cryptographic configuration.

3.2.5.3 Tests

ECDSA Algorithm Tests

ECDSA FIPS 186-4 Signature Generation Test

100 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

101 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

102 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.

103 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

104 For each modulus size/hash algorithm selected, the evaluator generates 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.

105 The evaluator verifies that the TOE confirms correct signatures on the original messages and detects the errors introduced in the altered messages.

Findings:	The vendor uses the CAVP certificate A2558 for ECDSA and RSA signature generation and verification. These are described in [ST] Table 4.

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

3.2.6.1 TSS

106 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:	[ST] Section 6.2.6 lists the parts of the TSF that implement the claimed hashing algorithms. These algorithms are consistent with the SFR claims. Each TSF function that uses a hash function is listed and includes the hashing algorithm(s) used by that component.

3.2.6.2 Guidance Documentation

107 The evaluator checks the AGD documents to determine that any configuration that is required to configure the required hash sizes is present.

Findings:	[AGD] Section 3.3 states that FIPS mode is enabled by default and no further configuration is needed to achieve the evaluated cryptographic configuration.

3.2.6.3 Tests

108 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.

109 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

110 The evaluators 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 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.

Short Messages Test - Byte-oriented Mode

111 The evaluators 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

112 The evaluators 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

113 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

114 This test is for byte-oriented implementations only. The evaluators 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 evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

Findings:	The vendor uses the CAVP certificate A2558 for SHA-1, SHA-256 and SHA-512 hashes. These are described in [ST] Table 4.

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

3.2.7.1 TSS

115 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:	[ST] Section 6.2.7 and table 14 lists the HMAC algorithms used in SSH connections. These algorithms are consistent with the SFR claims. The detail includes the block size, key size, and output digest size. The hash function used is directly implied by the HMAC algorithm.

3.2.7.2 Guidance Documentation

116 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:	[AGD] Section 3.3 states that FIPS mode is enabled by default and no further configuration is needed to achieve the evaluated cryptographic configuration.

3.2.7.3 Tests

117 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.

Findings:	The vendor uses the CAVP certificate A2558 for HMAC-SHA-256 and HMAC-SHA-512 keyed hashes. These are described in [ST] Table 4.

3.2.8 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

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

3.2.8.1 TSS

119 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:	[ST] Section 6.2.9 states “The TOE contains a CTR_DRBG that is seeded from a CPU provided entropy source. Entropy from the noise is conditioned and used to seed the DRBG with 256 bits of full entropy.”

3.2.8.2 Guidance Documentation

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

Findings:	[AGD] Section 3.3 states that FIPS mode is enabled by default and no further configuration is needed to achieve the evaluated cryptographic configuration.

3.2.8.3 Tests

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

122 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 verifies 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).

123 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 verifies 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.

124 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.

Findings:	The vendor uses the CAVP certificate A2558 for the DRBG. This is described in [ST] Table 4.

3.3 Identification and Authentication (FIA)

3.3.1 FIA_AFL.1 Authentication Failure Management

3.3.1.1 TSS

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

Findings:	The TOE only supports remote administration via SSH. [ST] Section 6.3.5 states that the TOE authentication failure threshold and time period are configurable. Once configured, a user that meets the threshold of successive unsuccessful attempts will be unable to login until the configured time period.

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

Findings:	[ST] Section 6.3.5 states that the local console does not implement the lockout mechanism.

3.3.1.2 Guidance Documentation

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

Findings:	[AGD] Section 3.7 points to the [ADMIN] “changing parameters” section which details how to change the number of successive unsuccessful authentication attempts prior to getting locked out and how to configure the lockout time period which is the only mechanism claimed in the ST.

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

Findings:	[AGD] Section 3.7 states that access is always maintained to the local CLI and no further configuration is needed.

3.3.1.3 Tests

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

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

High-Level Test Description
Using the SSH interface, set the login threshold to 3 attempts. Change the lockout duration to 30 seconds. Using the SSH interface, log into the TOE twice using an incorrect password. On the third attempt, log in correctly and verify that the threshold has not been reached. Using the SSH interface, log into the TOE three times using an incorrect password. On the fourth attempt, log in correctly and verify that the threshold has been reached and that the user cannot log in. After waiting the lockout duration period, successfully log in with the correct credentials.
Finding: Pass. The evaluator confirmed that the TOE did not allow the user to log on during the lockout duration after the incorrect login threshold was met.

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

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

Test Not Applicable	[ST] only claims time-based lockout.

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

High-Level Test Description
This is performed in conjunction with FIA_AFL.1 Test 1.
Finding: Pass. The evaluator confirmed that the TOE did not allow access after the login threshold was met but before the lockout duration was met. Once the lockout duration was exceeded, the TOE allowed the user to log in.

3.3.2 FIA_PMG_EXT.1 Password Management

3.3.2.1 TSS

130 The evaluator shall examine the TSS to determine that it contains the lists of the supported special character(s) and minimum and maximum number of charters supported for administrator passwords.

Findings:	[ST] Section 6.3.1 states the characters that passwords may be composed of. This is consistent with the claims in the FIA_PMG_EXT.1 SFR. The minimum password length is configurable by the Administrator from 15 to 1024 characters.

3.3.2.2 Guidance Documentation

131 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:	[AGD] Section 3.4 identifies the character set that password may be composed of as well as the minimum password length.

3.3.2.3 Tests

132 The evaluator shall perform the following tests.

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

High-Level Test Description
Set the minimum password length to 15 characters. Attempt to set a password less than the minimum length and show it is not accepted. Attempt to set passwords that fail to include characters from the out-of-the-box password complexity requirements and show they are not accepted. Attempt to set a password that meets the complexity and length requirements and show it is accepted. Show the password can be used on applicable management interfaces to log in successfully. Show that an admin with privileges can change another user's password and that the audit log reflects this capability.
Finding: Pass. The evaluator confirmed that the TOE accepts passwords that are comprised of the expected character set, that have the required minimum length, and that meet the required complexity.

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
This is performed in conjunction with FIA_PMG_EXT.1 Test 1.
Finding: Pass. The evaluator confirmed that the TOE does not accept passwords that do not meet the minimum length or complexity requirements.

3.3.3 FIA_UIA_EXT.1 User Identification and Authentication

3.3.3.1 TSS

133 The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon".

Findings:	[ST] Section 6.3.2 and [ST] Section 6.3.3 lists the available local and remote interfaces for a user to login to and the logon processes for them and what constitutes a "successful logon". Administrator credentials do not differ between the interfaces; thus the same credentials are used for both local and remote logon.

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

Findings:	[ST] Section 6.3.2 states the warning banner is displayed prior to authentication. No TOE administrative access is permitted until an administrator is successfully identified and authenticated.

135 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 and FIA_UAU_EXT.2, 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.

Findings:	The TOE is not a distributed TOE.

136 For distributed TOEs, the evaluator shall examine the TSS to determine that it describes for each TOE component which actions are allowed before user identification and authentication. The description shall cover authentication and identification for local and remote TOE administration. For each TOE component that does not support authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 the TSS shall describe any unauthenticated services/services that are supported by the component.

Findings:	The TOE is not a distributed TOE.

3.3.3.2 Guidance Documentation

137 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:	[AGD] Section 3.7 refers to the [ADMIN] sections “Authenticate with a key” and “Create new SSH host-keys” to detail the steps needed to login to the TOE via SSH public/private key. This section also refers to [ADMIN] section “Change password” to configure the administrator password.

3.3.3.3 Tests

138 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 that credential/login method, 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
Log into the identified management interface using a known-good credential and logout. Login into the identified management interface using a known-bad credential and verify that the login is not successful. Ensure the appropriate audit messages appear.

High-Level Test Description
Finding: Pass. The evaluator confirmed that providing the correct credentials resulted in the TOE granting access. The evaluator also confirmed that providing incorrect credentials resulted in the TOE denying access.

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

High-Level Test Description
The TOE does not have any services configured prior to I&A other than a TOE banner. Using the remote SSH interface, initiate a logon attempt to the TOE and verify that only the TOE banner is displayed prior to log in.
Finding: Pass. The evaluator confirmed that the TOE displayed the warning banner prior to logging in and that the user must log onto the TOE to access any other services.

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
The TOE does not have any services configured prior to I&A other than the TOE banner. Using the local CLI, initiate a logon attempt to the TOE and verify that only the TOE banner is displayed prior to log in.
Finding: Pass. The evaluator confirmed that the TOE displayed the warning banner prior to logging in and that the user must log onto the TOE to access any other services.

d. Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.

Test Not Applicable	The TOE is not a distributed TOE.

3.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

139 Evaluation Activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.

3.3.5 FIA_UAU.7 Protected Authentication Feedback

3.3.5.1 TSS

140 None

3.3.5.2 Guidance Documentation

141 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:	The TOE provides obscured feedback of passwords by default; no further steps need to be taken.

3.3.5.3 Tests

142 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
Log into the local management interface. Ensure the password field does not echo characters – even a masking character -- as claimed by the ST.
Finding: Pass. The evaluator confirmed that the TOE does not echo any characters when the password is typed in the local console.

3.4 Security management (FMT)

3.4.1 General requirements for distributed TOEs

3.4.1.1 TSS

143 For distributed TOEs it is required to verify the TSS to ensure that it 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.

Findings:	The TOE is not a distributed TOE.

3.4.1.2 Guidance Documentation

144 For distributed TOEs it is required to verify 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.

Findings:	The TOE is not a distributed TOE.

3.4.1.3 Tests

145 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 a distributed TOE.

3.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

3.4.2.1 TSS

146 For distributed TOEs see [SD] chapter 2.4.1.1. There are no specific requirements for non-distributed TOEs.

Findings:	The TOE is not a distributed TOE.

3.4.2.2 Guidance Documentation

147 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:	[AGD] Section 2.4 provides instructions on how to perform a manual update.

148 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).

Findings:	The TOE is not a distributed TOE.

3.4.2.3 Tests

149 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.

150 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.

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.
Finding: Pass. The evaluator confirmed that the TOE does not allow TOE updates to be initiated prior to administrator logon.

3.4.3 FMT_MTD.1/CoreData Management of TSF Data

3.4.3.1 TSS

151 The evaluator shall examine the TSS to determine that, for each administrative function identified in the guidance documentation; those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

Findings:	[ST] Section 6.4.4 states "Users are required to login before being provided with access to any administrative functions."

152 If 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.

Findings:	N/A because the TOE does not support handling of X.509v3 certificates.

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:	[AGD] Section 3 contains configuration information for an administrator to manipulate the TSF. The specific sections for each SFR are addressed in the remaining Guidance Documentation Assurance Activities within this document. Additionally, [ADMIN] Section 3.5.1 "USB Console" and 3.5.4 "SSH Server" state there is a singular user "admin" that has access to 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.

Findings:	N/A because the TOE does not support handling of x.509v3 certificates.

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.

Findings:	No separate testing is needed for this SFR. Testing of the administrator's ability to exercise TOE management is covered throughout the test assurance activities in this document. Additionally, FMT_MOF.1/Functions, FMT_MOF.1/Services and FMT_MTD.1/CryptoKeys testing shows that a non-administrator cannot exercise any TOE management.

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:	[ST] Section 6.4.7 lists management capabilities of the TOE. All functions are available via local and remote administration. Additionally, throughout testing the evaluator confirmed that all claimed management functions 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:	[ST] Section 6.3.2 describes the local administrative interface as and Administrative CLI via direct serial connection. [AGD] Section 1.3.3 states that the CLI is accessed by a direct serial connection.

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.

Findings:	The TOE is not a distributed TOE.

3.4.4.2 Guidance Documentation

160 See [SD] section 2.4.4.1.

Findings:	This is covered by the findings in section 3.4.4.1 above.

3.4.4.3 Tests

161 The evaluator tests management functions as part of testing the SFRs identified in section 4.4.4. 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.

Findings:	The evaluator tested management functions as part of testing the SFRs. No separate testing for FMT_SMF.1 is required unless one of the management functions in FMT_SMF.1.1 has not already been exercised under any other SFR.

3.4.5 FMT_SMR.2 Restrictions on security roles

3.4.5.1 TSS

162 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.

Findings:	[ST] Section 6.4.5 states all user accounts are Security Administrators. A single 'Admin' role is defined which can be accessed via SSH or local serial connection. Only Security Administrators can manage TSF data.

3.4.5.2 Guidance Documentation

163 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:	[AGD] Section 3.2 "Administration Interfaces" outlines the two methods of administering the TOE—via local CLI and CLI over SSH. This section also states how public key authentication can be configured.

3.4.5.3 Tests

164 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; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.

Findings:	There are no explicit test activities and therefore none are recorded here. All interfaces are tested throughout testing.

3.5 Protection of the TSF (FPT)

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

3.5.1.1 TSS

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

Findings:	[ST] Section 6.5.1 lists types of keys and how each is stored. All keys are stored in plaintext and cannot be viewed through an interface specifically designed for that purpose.

3.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

3.5.2.1 TSS

166 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:	[ST] Section 6.5.2 states passwords are hashed using SHA-512 and stored in flash memory. This section also states that passwords cannot be viewed through an interface designed specifically for that purpose.

3.5.3 FPT_TST_EXT.1 TSF testing

3.5.3.1 TSS

167 The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF; 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.

Findings:	[ST] Section 6.5.3 states the tests that are performed at start-up . These tests include Firmware integrity tests, OpenSCAP tests, and the FIPS test suite.The purpose of these tests is to insure correct operation of cryptographic functionality, the FIPS module and the correct TOE image. If any of the tests fail, the functionality will not be available.

168 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.

Findings:	The TOE is not a distributed TOE.

3.5.3.2 Guidance Documentation

169 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:	[AGD] Section 2.3 states that a failure may occur and if it does, the administrator should reinstall the TOE. If the failure continues, then the administrator should contact the vendor for support. This is consistent with [ST] section 6.5.3 since it describes that the TOE operation will be unavailable in the event of a failure.

170 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.

Findings:	The TOE is not a distributed TOE.

3.5.3.3 Tests

171 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.

172 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.

173 The evaluator shall either verify that the self-tests described above are carried out during initial start-up or that the developer has justified any deviation from this.

High-Level Test Description
Reset the TOE and witness that the startup includes an indicator that self-tests were executed and passed permitting the device to operate.
Finding: Pass. The evaluator confirmed that the TOE indicates that the self-tests ran successfully during startup via console output and the audit trail.

174 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 a distributed TOE.

3.5.4 FPT_TUD_EXT.1 Trusted Update

3.5.4.1 TSS

175 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 needs to describe how and when the inactive version becomes active. The evaluator shall verify this description.

Findings:	[ST] Section 6.5.4 states that the current firmware version may be queried using any administrative interface (SSH or local CLI). The [ST] does not claim delayed activation.

176 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. Alternatively an approach using a published hash can be used. In this case the TSS shall detail this mechanism instead of the digital signature verification mechanism. The evaluator shall verify that the TSS describes the method by which the digital signature or published hash is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature or published hash of the update, and the actions that take place for both successful and unsuccessful signature verification or published hash verification.

Findings:	[ST] Section 6.5.4 states the Security Administrator manually initiates TOE updates from the Bash CLI. The TOE implements update verification via digital signature (RSA). If the update succeeds the TOE is rebooted to the new version. If the update fails the update is aborted and an error message is displayed. Updates are obtained via physical delivery.

177 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.

Findings:	Both ‘support automatic checking for updates’ and ‘support automatic updates’ are not chosen for the selection in FPT_TUD_EXT.1.2.

178 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 should examine the guidance documentation instead.

Findings:	The TOE is not a distributed TOE.

179 If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the TSS contains a description of how the certificates are contained on the device. The evaluator also ensures that the TSS (or guidance documentation) describes how the certificates are installed/updated/selected, if necessary.

Findings:	[ST] Section 6.5.4 states the signature is verified using a hardcoded public key.

180 If a published hash is used to protect the trusted update mechanism, then the evaluator shall verify that the trusted update mechanism does involve an active authorization step of the Security Administrator, and that download of the published hash value, hash comparison and update is not a fully automated process involving no active authorization by the Security Administrator. In particular, authentication as Security Administration according to FMT_MOF.1/ManualUpdate needs to be part of the update process when using published hashes.

Findings:	A published hash is not used to verify updates.

3.5.4.2 Guidance Documentation

181 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 needs to describe how to query the loaded but inactive version.

Findings:	[AGD] Section 2.2 “Verifying the TOE” shows the command to query the current version of the software. Delayed activation is not claimed.

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

Findings:	[AGD] “Trusted Update” bullet point in section 1.3.3 says “The TOE ensures the authenticity and integrity of software updates via digital signature.” [ADMIN] Section “Image files available on the TOE” shows examples of the digital signature check failing and passing.

183 If a published hash is used to protect the trusted update mechanism, the evaluator shall verify that the guidance documentation describes how the Security Administrator can obtain authentic published hash values for the updates.

Findings:	A published hash is not used for the trusted update.

184 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 user; it does not need to give information about the internal communication that takes place when applying updates.

Findings:	The TOE is not a distributed TOE.

185 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.

Findings:	The TOE is not a distributed TOE.

186 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 also ensures that the Guidance Documentation describes how the certificates are installed/updated/selected, if necessary.

Findings:	Certificate based update authentication is not claimed.

3.5.4.3 Tests

187 The evaluator shall perform the following tests:

a. Test 1: The evaluator performs 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 verifies 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 performs the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again.

High-Level Test Description
Get the current version of the TOE. Attempt to install a legitimate update of the TOE. After the install, get the current version of the TOE and ensure it is consistent with the newly installed version.
Finding: Pass. The evaluator attempted to upgrade the TOE to a known good build and confirmed that the TOE successfully upgraded to the new build.

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 first confirms that no updates are pending and then performs 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 obtains or produces illegitimate updates as defined below and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates:

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

2) An image that has not been signed

3) 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)

4) If the TOE allows a delayed activation of updates the TOE must be able to display both the currently executing version and most recently installed version. 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, 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 bad image, an unsigned image and a badly signed image for upgrade. After each attempt, get the current version of the TOE using all available means and ensure they are consistent.
Finding: Pass. The evaluator attempted to upgrade the TOE separately with an upgrade that was modified, with an unsigned and with an upgrade with a modified signature and confirmed that the TOE rejected these builds and did not install them.

c. Test 3 [conditional]: If the TOE itself verifies a hash value over an image against a published hash value (i.e. reference value) that has been imported to the TOE from outside such that the TOE itself authorizes the installation of an image to update the TOE, the following test shall be performed (otherwise the test shall be omitted. If the published hash is provided to the TOE by the Security Administrator and the verification of the hash value over the update file(s) against the published hash is performed by the TOE, then the evaluator shall perform the following tests. The evaluator first confirms that no update is pending and then performs 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.

1) The evaluator obtains or produces an illegitimate update such that the hash of the update does not match the published hash. The evaluator provides the published hash value to the TOE and calculates the hash of the update either on the TOE itself (if that functionality is provided by the TOE), or else outside the TOE. The evaluator confirms that the hash values are different, and attempts to install the update on the TOE, verifying that this fails because of the difference in hash values (and that the failure is logged). Depending on the implementation of the TOE, the TOE might not allow the user to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE

2) The evaluator uses a legitimate update and tries to perform verification of the hash value without providing the published hash value to the TOE. The evaluator confirms that this attempt fails. The evaluator confirms that this attempt fails. Depending on the implementation of the TOE it might not be possible to attempt the verification of the hash value without providing a hash value to the

TOE, e.g. if the hash value needs to be handed over to the TOE as a parameter in a command line message and the syntax check of the command prevents the execution of the command without providing a hash value. In that case the mechanism that prevents the execution of this check shall be tested accordingly, e.g. that the syntax check rejects the command without providing a hash value, and the rejection of the attempt is regarded as sufficient verification of the correct behaviour of the TOE in failing to verify the hash. The evaluator then attempts to install the update on the TOE (in spite of the unsuccessful hash verification) and confirms that this fails. Depending on the implementation of the TOE, the TOE might not allow to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE

3) If the TOE allows delayed activation of updates, the TOE must be able to display both the currently executing version and most recently installed version. The handling of version information of the most recently installed version might differ between different TOEs. Depending on the point in time when the attempted update is rejected, the most recently installed version might or might not be updated. 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, current version and most recently installed version, reflect the same version information as prior to the update attempt.

Test Not Applicable	The TOE does not support published hashes.

188 If the verification of the hash value over the update file(s) against the published hash is not performed by the TOE, Test 3 shall be skipped.

189 The evaluator shall perform Test 1, Test 2 and Test 3 (if applicable) for all methods supported (manual updates, automatic checking for updates, automatic updates).

190 For distributed TOEs the evaluator shall perform Test 1, Test 2 and Test 3 (if applicable) for all TOE components.

3.5.5 FPT_STM_EXT.1 Reliable Time Stamps

3.5.5.1 TSS

191 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:	[ST] Section 6.5.5 states that the TOE makes use of secure NTP to maintain date and time. At least 3 time servers can be configured using SHA1 pre-shared keys are used to maintain reliability. The TOE makes use of time for audit record timestamps, session timeouts and lockout enforcement.

192 **[NIAP TD0632]** 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.

Not Applicable:	This is not applicable since the [ST] does not make the “obtain time from the underlying virtualization system” selection.

3.5.5.2 Guidance Documentation

193 The evaluator examines 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:	[AGD] Section 3.5 refers to [ADMIN] section “Add/Remove NTP servers” outlines how to add an NTP server on the TOE.

194 **[NIAP TD0632]** 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.

Not Applicable:	This is not applicable since the TOE does not support obtaining time from an underlying VS.

3.5.5.3 Tests

195 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 uses the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

Test Not Applicable	The TOE does not support direct setting of time by a security administrator.

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 will 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.

High-Level Test Description
Configure the TOE to connect with an NTP server and verify that the time successfully syncs.
Finding: Pass. The evaluator configured the TOE to sync with NTP and confirmed that the TOE successfully updates its time from the NTP server.

196 **[NIAP TD0632]** c. Test 3: [conditional] If the TOE obtains time from the underlying VS, the evaluator shall record the time on the TOE, modify the time on the underlying VS, and verify the modified time is reflected by the TOE. If there is a delay between the setting the time on the VS and when the time is reflected on the TOE, the evaluator shall ensure this delay is consistent with the TSS and Guidance.

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

197 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 TOE does not support independent time information.

3.6 TOE Access (FTA)

3.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

3.6.1.1 TSS

198 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:	[ST] Section 6.6.1 states that the Security Administrator may configure the TOE to terminate an inactive local interactive session following a specified period of time.

3.6.1.2 Guidance Documentation

199 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:	[ADMIN] Section “Changing parameters” and [AGD] section 3.2 “Administration interfaces” outlines how to change the inactivity timeout. [AGD] Section 3.2 also states how local or remote sessions are terminated.

3.6.1.3 Tests

200 The evaluator shall perform the following test:

a. Test 1: The evaluator follows the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.

High-Level Test Description
For each of 1, 3, 5 minutes: Change the idle timeout to this value; Log into the device; Wait for the full duration of the timeout. The session should terminate.

High-Level Test Description
Finding: Pass. The evaluator configured the TOE's idle timeout value and logged into the TOE. The evaluator waited until the timeout threshold was met and confirmed the TOE terminated the session. The evaluator performed this test for timeout values of 1, 3 and 5 minutes.

3.6.2 FTA_SSL.3 TSF-initiated Termination

3.6.2.1 TSS

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

Findings:	[ST] Section 6.6.2 states that the Security Administrator may configure the TOE to terminate an inactive remote interactive session following a specified period of time.

3.6.2.2 Guidance Documentation

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

Findings:	[ADMIN] Section “Changing parameters” and [AGD] section 3.2 “Administration interfaces” outlines how to change the inactivity timeout for both CLI/Console and CLI/SSH.

3.6.2.3 Tests

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

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

High-Level Test Description
For each of 1, 3, 5 minutes: Change the idle timeout to this value; Log into the device; Wait for the full duration of the timeout without sending any keep alives. The session should terminate.
Finding: Pass. The evaluator configured the TOE's idle timeout value and logged into the TOE. The evaluator waited until the timeout threshold was met and confirmed the TOE terminated the session. The evaluator performed this test for timeout values of 1, 3 and 5 minutes.

3.6.3 FTA_SSL.4 User-initiated Termination

3.6.3.1 TSS

204 The evaluator shall examine the TSS to determine that it details how the local and remote administrative sessions are terminated.

Findings:	[ST] Section 6.6.3 states that Administrative users may terminate their own sessions by using the “exit” command.

3.6.3.2 Guidance Documentation

205 The evaluator shall confirm that the guidance documentation states how to terminate a local or remote interactive session.

Findings:	[AGD] Section 3.2 “Administration interfaces” says “Local and remote sessions are terminated using, exit.”

3.6.3.3 Tests

206 For each method of remote administration, the evaluator shall perform the following tests:

a. Test 1: The evaluator initiates an interactive local session with the TOE. The evaluator then follows 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 Log out using the TSFI previous discussed. Verify that the session has been terminated.
Finding: Pass. The evaluator logged into the TOE’s serial console then initiated a logout. The evaluator confirmed that the TOE terminated the session with the user.

b. Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows 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. Log out using the TSFI previous discussed. Verify that the session has been terminated.
Finding: Pass. The evaluator logged into the TOE via SSH then initiated a logout. The evaluator confirmed that the TOE terminated the session with the user.

3.6.4 FTA_TAB.1 Default TOE Access Banners

3.6.4.1 TSS

207 The evaluator shall check the TSS to ensure that it details each administrative method of access (local and 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:	[ST] Section 6.3.2 contains the methods of access to the TOE: CLI via direct serial connection and CLI via SSH. [ST] Section 6.6.4 states "The TOE displays an administrator configurable message to users prior to login at the CLI."

3.6.4.2 Guidance Documentation

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

Findings:	[AGD] Section 3.2 "Administration interfaces" and [ADMIN] section "Changing the login banner" describe how to configure the banner message"

3.6.4.3 Tests

209 The evaluator shall also perform the following test:

a. Test 1: The evaluator follows 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
Log into the SSH CLI interface. Change the banner to a random string. Log into fresh sessions for all interactive interfaces and show that the banner was modified and is presented prior to I&A.
Finding: Pass. The evaluator logged into the TOE and changed the banner message. The evaluator then confirmed that the new message is displayed before logging into the local console and remotely over SSH.

3.7 Trusted path/channels (FTP)

3.7.1 FTP_ITC.1 Inter-TSF trusted channel

3.7.1.1 TSS

210 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:	[ST] Section 6.7.1 states that the TOE supports secure communication with an audit server as a client via SSH per FCS_SSHC_EXT.1 and lists the algorithms supported by the TOE.

3.7.1.2 Guidance Documentation

211 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:	[AGD] Section 3.6 contains instructions to enable remote syslog logging. If the connection is broken, the TOE will automatically attempt to reconnect. If this problem persists, section 3.6 instructs the administrator to stop and start the audit function.

3.7.1.3 Tests

212 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.

213 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.

High-Level Test Description
The TOE maintains trusted channels to the remote audit log, which are set up as per the evaluated configuration. Test 2 below demonstrates the successful execution of this test.
Finding: Pass. The evaluator confirmed in conjunction with Test 2 below that the TOE successfully communicates with the remote audit server.

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.

High-Level Test Description
Engage wireshark over the appropriate interface. Log into the CLI and disable and re-enable the logging interface. Examine wireshark and verify that the log interface initiates an SSH connection to the remote logging server and the communications are not in plaintext.
Finding: Pass. The evaluator enabled the remote logging connection on the TOE and confirmed that the TOE initiates a connection to the remote logging server and the connection successfully communicates via encrypted SSH.

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

Finding: Pass. The evaluator confirmed that channel data is not sent in plaintext in conjunction with the previous test case.

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, physically 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 physical 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 when the cable is removed from the device, another physical network device (e.g. a core switch) shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall not be performed at the virtual node (e.g. virtual switch) and must be physical in nature.

High-Level Test Description
Engage wireshark over the logging interface. Physically disconnect the remote logging server (disconnect from the remote end rather that from the TOE end to ensure that the TOE is unable to invoke any layer 2 carrier-sensing mechanism). Wait 15 seconds. Physically reconnect the remote logging server. Examine wireshark and verify that the log interface continues to send encrypted Application Data packets. Repeat the above with a 15 minute timeout.
Finding: Pass. The evaluator confirmed an encrypted SSH connection to the remote logging server then disconnected the physical connection for 15 seconds before reconnecting. Upon restoring the physical connection, the evaluator confirmed that the TOE continues to communicate with the remote server via encrypted SSH. The evaluator then repeated this test with a 15 minute disconnect and observed the same result.

Further assurance activities are associated with the specific protocols.

214 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 a distributed TOE.

215 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

216 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:	[ST] Section 6.7.2 states that the TOE provides an Administrative CLI via SSH per FCS_SSHS_EXT.1 as a trusted path for remote administration.

3.7.2.2 Guidance Documentation

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

Findings:	[ADMIN] Section “SSH server” outlines the SSH server provided for remote administrative sessions. [ADMIN] Section “Authenticate with a key” provides instructions on allowing login via public/private keys.

3.7.2.3 Tests

218 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.

High-Level Test Description
The only trusted path is SSH, which is set up as per the evaluated configuration. It is constantly tested throughout the evaluation.
Finding: Pass. The evaluator confirmed in conjunction with Test 2 below that the TOE successfully communicates via encrypted SSH for remote administration.

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

High-Level Test Description
Engage wireshark over the appropriate interface. Log into the trusted path. Examine wireshark and verify that the trusted path sends encrypted traffic after any initial plaintext protocol negotiation occurs.
Finding: Pass. The evaluator connected to the TOE as a remote administrator over SSH and confirmed that the communication is encrypted.

219 Further assurance activities are associated with the specific protocols.

220 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 a distributed TOE.

4 Evaluation Activities for Optional Requirements

221 No optional requirements have been selected by this evaluation.

5 Evaluation Activities for Selection-Based Requirements

5.1 Cryptographic Support (FCS)

5.1.1 FCS_NTP_EXT.1 NTP Protocol

5.1.1.1 TSS

FCS_NTP_EXT.1.1

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

Findings:	[ST] Section 6.2.8 states that the TOE uses NTPv4 and SHA-1 pre-shared keys for authentication.

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

Findings:	[ST] Section 6.2.8 states that the TOE supports NTPv4 using SHA-1 authentication.

5.1.1.2 Guidance Documentation

FCS_NTP_EXT.1.1

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

Findings:	The TOE only claims one NTP version (NTPv4), [ADMIN] section "Add/Remove NTP server" section contains instructions on adding NTP servers.

FCS_NTP_EXT.1.2

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

Findings:	The TOE only claims one algorithm (SHA1), so no configuration is necessary.

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

Findings:	The TOE only claims one algorithm (SHA1), so no configuration is necessary.

FCS_NTP_EXT.1.3

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

Findings:	[AGD] Section 3.5 states that no additional configuration is needed to reject broadcast or multicast time updates.

5.1.1.3 Tests

FCS_NTP_EXT.1.1

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

High-Level Test Description
Allow the TOE to poll the test machine for the current time using NTP and ensure the NTP version matches the ST.
Finding: Pass. The evaluator successfully communicated from the TOE with the NTP server and confirmed that the TOE uses NTPv4.

FCS_NTP_EXT.1.2

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

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

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

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

High-Level Test Description
Attempt to establish a working NTP connection between the TOE and the NTP server. Show that the connection is using secure NTP. Change the digest algorithm on the server to not match what the TOE supports and show the time no longer gets synched on the TOE.

High-Level Test Description
Finding: Pass. The evaluator confirm that the TOE syncs with the NTP server using SHA1. The evaluator then configured the NTP server to only use MD5 and confirmed that the TOE does not sync with the NTP server since MD5 is not a supported message digest algorithm.

FCS_NTP_EXT.1.3

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

High-Level Test Description
Configure an NTP server to send NTP packets to broadcast and multicast addresses and ensure the TOE does not accept these packets and sync the time to them.
Finding: Pass. The evaluator configured the NTP server to send time updates to broadcast and multicast addresses. The evaluator confirmed that when the TOE receives these time updates, the TOE does not sync its time with the NTP server.

FCS_NTP_EXT.1.4

234 **[NIAP TD0528]** Test 1: The evaluator shall confirm the TOE supports configuration of at least three (3) NTP time sources. The evaluator shall configure at least three NTP servers to support periodic time updates to the TOE. The evaluator shall confirm the TOE is configured to accept NTP packets that would result in the timestamp being updated from each of the NTP servers. The evaluator shall check that the time stamp is updated after receipt of the NTP packets. The purpose of this test to verify that the TOE can be configured to synchronize with multiple NTP servers. It is up to the evaluator to determine that the multi- source update of the time information is appropriate and consistent with the behaviour prescribed by the RFC 1305 for NTPv3 and RFC 5905 for NTPv4.

High-Level Test Description
The evaluator added 3 time servers and ensured the TOE syncs with each one, note that the position of the time server is independent of the capability to acquire the time, NTP packets are transmitted from each of the three time servers.
Finding: Pass. The evaluator configured the TOE with 3 time servers and confirmed that the TOE communicates with each independent time server.

235 **[NIAP TD0528]** Test 2: (The intent of this test is to ensure that the TOE would only accept NTP updates from configured NTP Servers).

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

unsolicited updates in a way that would be consistent with the behaviour of a correctly-functioning NTP server.

High-Level Test Description
Use the lightship ntpd tool to spoof the src IP of the server response to the TOEs NTP client request and ensure the TOE does not accept the time change.
Finding: Pass. The evaluator configured an NTP server to send otherwise valid time updates to the TOE from a spoofed IP address and confirmed that the TOE does not update its time.

5.1.2 FCS_SSHC_EXT.1 SSH Client

5.1.2.1 TSS

FCS_SSHC_EXT.1.2

237 **[NIAP TD0636]** The evaluator shall check to ensure that the TSS contains a list of the public key algorithms that are acceptable for use for user authentication and that this list is consistent with asymmetric key generation algorithms selected in FCS_CKM.1, hashing algorithms selected in FCS_COP.1/Hash, and signature generation algorithms selected in FCS_COP.1/SigGen. The evaluator shall confirm the TSS is unambiguous in declaring the TOE's ability to authenticate itself to a remote endpoint with a user-based public key.

Findings:	[ST] Section 6.2.10 states that the TOE SSH client supports public key authentication using rsa-sha2-512 at 2048, 3072, and 4096 key sizes for user keys. This is consistent with the claims in FCS_CKM.1, FCS_COP.1/Hash and FCS_COP.1/SigGen.

238 **[NIAP TD0636]** If password-based authentication method has been selected in the FCS_SSHC_EXT.1.2, then the evaluator shall confirm it is also described in the TSS.

Findings:	Password based authentication methods have not been selected in the [ST].

FCS_SSHC_EXT.1.3

239 The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled.

Findings:	[ST] Section 6.2.10 states that the TOE automatically drops packets greater than 256KB.

FCS_SSHC_EXT.1.4

240 The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

Findings:	[ST] Section 6.2.10 states that the TOE uses AES-CTR-128 and AES-CTR-256 for SSH encryption which is consistent with those claimed in FCS_SSHC_EXT.1.4.

FCS_SSHC_EXT.1.5

241 **[NIAP TD0636]** The evaluator shall confirm the TSS describes how a host-key public key (i.e., SSH server's public key) is associated with the server identity.

Findings:	[ST] Section 6.2.10 states that the TOE authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key.

242 **[NIAP TD0636]** The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the host-key public key algorithms supported by the TOE are specified as well. The evaluator shall check the TSS to ensure that the host-key public key algorithms specified are identical to those listed for this component.

Findings:	[ST] Section 6.2.10 states that the TOE SSH client supports host key authentication using ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521. This is consistent with the claims in FCS_SSHC_EXT.1.5.

243 If x509v3-based public key authentication algorithms are claimed, the evaluator shall confirm that the TSS includes the description of how the TOE establishes the server's identity and how this identity is confirmed with the one that is presented in the provided certificate. For example, the TOE could verify that a server's configured IP address matches the one presented in the server's x.509v3 certificate.

Findings:	The ST does not claim x509v3-based public key authentication algorithms.

FCS_SSHC_EXT.1.6

244 The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component.

Findings:	[ST] Section 6.2.10 states that the TOE provides data integrity for SSH connections via hmac-sha2-256 and hmac-sha2-512 which is consistent with FCS_SSHC_EXT.1.6.

FCS_SSHC_EXT.1.7

245 The evaluator shall check the TSS to ensure that it lists the supported key exchange algorithms, and that that list corresponds to the list in this component.

Findings:	[ST] Section 6.2.10 states that the TOE supports diffie-hellman-group14-sha1, ecdh-sha2-nistp256, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521 which is consistent with FCS_SSHC_EXT.1.7.

FCS_SSHC_EXT.1.8

246 The evaluator shall check that the TSS specifies the following:

1. Both thresholds are checked by the TOE.
2. Rekeying is performed upon reaching the threshold that is hit first.

Findings:	[ST] Section 6.2.10 states that the TOE will re-key SSH connections after 1 hour or after an aggregate of 1 gig of data has been exchanged (whichever occurs first).

5.1.2.2 Guidance Documentation

FCS_SSHC_EXT.1.2

247 **[NIAP TD0636]** 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 initiated by the TOE.

Findings:	[AGD] Section 3.8 provides instructions for how the administrator can add a trusted syslog server to the known hosts file and configure the TOE configure the SSH connection for remote logging to ensure that the TOE only initiates communications to trusted syslog servers.

FCS_SSHC_EXT.1.4

248 The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Findings:	[AGD] Section 3.8 states that no additional configuration is needed for the TOE SSH client connection.

FCS_SSHC_EXT.1.5

249 The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Findings:	[AGD] Section 3.8 states that no additional configuration is needed for the TOE SSH client connection.

FCS_SSHC_EXT.1.6

250 The evaluator shall also check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed).

Findings:	[AGD] Section 3.8 states that no additional configuration is needed for the TOE SSH client connection.

FCS_SSHC_EXT.1.7

251 The evaluator shall also check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections with the TOE.

Findings:	[AGD] Section 3.8 states that no additional configuration is needed for the TOE SSH client connection.

FCS_SSHC_EXT.1.8

252 If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, then the evaluator shall check that the guidance documentation describes how to configure those thresholds. Either the allowed values are specified in the guidance documentation and must not exceed the limits specified in the SFR (one hour of session time, one gigabyte of transmitted traffic) or the TOE must not accept values beyond the limits specified in the SFR. The evaluator shall check that the guidance documentation describes that the TOE reacts to the first threshold reached.

Findings:	[AGD] Section 3.8 states that no additional configuration is needed for the TOE SSH client connection.

5.1.2.3 Tests

FCS_SSHC_EXT.1.2

253 **[NIAP TD0636]** Test objective: The purpose of these tests is to check the authentication of the client to the server using each claimed authentication method.

254 Test 1: For each claimed public-key authentication method, the evaluator shall configure the TOE to present a public key corresponding to that authentication method (e.g., 2048-bit RSA key when using ssh-rsa public key). The evaluator shall establish sufficient separate SSH connections with an appropriately configured remote non-TOE SSH server to demonstrate the use of all claimed public key algorithms. It is sufficient to observe the successful completion of the SSH Authentication Protocol to satisfy the intent of this test.

High-Level Test Description
Configure all claimed public keys on the TOE and connect to an SSH server using each of the configured keys.
Finding: Pass. The evaluator attempted separate connections from the TOE to an SSH server using rsa-sha2-512 with public key sizes 2048, 3072, and 4096 and confirmed the communication was successful.

255 Test 2: [Conditional] If password-based authentication method has been selected in the FCS_SSHC_EXT.1.2, then following the guidance documentation the evaluator shall configure the TOE to perform password-based authentication with a remote SSH server to demonstrate that the TOE can successfully authenticate using a password as an authentication method.

Test Not Applicable	Password based authentication is not selected in the ST.

FCS_SSHC_EXT.1.3

256 The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

High-Level Test Description
Using a custom server tool, force the TOE to receive a packet larger than the expected TOE buffer size and show that the TOE rejects the packet in some way.
Finding: Pass. The evaluator established an SSH connection from the TOE then sent a packet larger than 256 kilobytes to the TOE and confirmed that the TOE rejected the packet and closed the connection upon receipt of the packet.

FCS_SSHC_EXT.1.4

257 The evaluator must ensure that only claimed ciphers and cryptographic primitives are used to establish a SSH connection. To verify this, the evaluator shall start session establishment for a SSH connection with a remote server (referred to as 'remote endpoint' below). 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 all the ciphers defined in the TSS for the TOE for SSH sessions, but no additional ones compared to the definition in the TSS. The

evaluator shall perform one successful negotiation of an SSH session to verify that the TOE behaves as expected. It is sufficient to observe the successful negotiation of the session to satisfy the intent of the test. If the evaluator detects that not all ciphers defined in the TSS for SSH are supported by the TOE and/or the TOE supports one or more additional ciphers not defined in the TSS for SSH, the test shall be regarded as failed.

High-Level Test Description
Permit the TOE client to connect to a test SSH server and capture the TOE client's advertised supported cipher algorithms. Verify that the advertised set matches the claimed set. Forcibly use an SSH server to permit connections from the TOE client using only one of those claimed ciphers and show that the connection is successful.
Finding: Pass. The evaluator established an SSH connection from the TOE and verified that the TOE supports aes128-ctr and aes256-ctr as specified by [ST].

FCS_SSHC_EXT.1.5

258 Test 1: The evaluator shall establish a SSH connection using each of the public key algorithms specified by the requirement to authenticate an SSH server to the TOE. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. Test objective: The purpose of this positive test is to check the authentication of the server by the client (when establishing the transport layer connection), and not for checking generation of the authentication message from the client (in the User Authentication Protocol). The evaluator shall therefore establish sufficient separate SSH connections (with an appropriately configured server) to cause the TOE to demonstrate use of all public key algorithms claimed in FCS_SSHC_EXT.1.5 in the ST.

High-Level Test Description
Use the TOE client and connect to a test SSH server which only provides a host key using the specified public key algorithms in turn. Show that the client authenticates the host.
Finding: Pass. The evaluator attempted separate connections from the TOE to the SSH server offering ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521 host keys and confirmed that the TOE successfully connected for each algorithm.

259 Test 2: The evaluator shall configure an SSH server to only allow a public key algorithm that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection from the TOE to the SSH server and observe that the connection is rejected.

High-Level Test Description
Ensure the TOE has a supported public/private key pair. Off-TOE, use a different public key (generated with the same public key algorithm). Permit the TOE client to connect to the non-TOE server. The connection attempt should fail.
Finding: Pass. The evaluator attempted an SSH connection from the TOE to the SSH server offering an ssh-dss host key and verified that the TOE rejected the connection upon receipt of the unsupported host key.

FCS_SSHC_EXT.1.6

260 Test 1: (conditional, if an HMAC or AEAD_AES_*_GCM algorithm is selected in the ST) The evaluator shall establish an SSH connection using each of the algorithms, except “implicit”, specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

261 Note: To ensure the observed algorithm is used, the evaluator shall ensure a non-aes*-gcm@openssh.com encryption algorithm is negotiated while performing this test.

High-Level Test Description
Using an SSH Server, forcibly permit only the claimed integrity algorithms and show that connections by the TOE SSH client are accepted to form a successful connection.
Finding: Pass. The evaluator attempted separate SSH connections from the TOE to the SSH server using hmac-sha2-256 and hmac-sha2-512 and confirmed the TOE successfully connected using each algorithm.

262 Test 2: (conditional, if an HMAC or AEAD_AES_*_GCM algorithm is selected in the ST) The evaluator shall configure an SSH server to only allow a MAC algorithm that is not included in the ST selection. The evaluator shall attempt to connect from the TOE to the SSH server and observe that the attempt fails.

263 Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*-gcm@openssh.com encryption algorithm is negotiated while performing this test .

High-Level Test Description
Using an SSH Server, forcibly permit an integrity algorithm which is not claimed by the TOE and show that a TOE SSH client connection results in a failed connection.
Finding: Pass. The evaluator attempted an SSH connection from the TOE to the SSH server only offering hmac-sha1 and confirmed that the TOE rejected the connection.

FCS_SSHC_EXT.1.7

264 Test 1: The evaluator shall configure an SSH server to permit all allowed key exchange methods. The evaluator shall attempt to connect from the TOE to the SSH server using each allowed key exchange method, and observe that each attempt succeeds.

High-Level Test Description
Using an SSH server, forcibly permit only one claimed key exchange mechanism at a time and show that the TOE client will successfully connect using that algorithm.
Finding: Pass. The evaluator attempted separate SSH connections from the TOE to the SSH server using diffie-hellman-group14-sha1, ecdh-sha2-nistp256, diffie-hellman-group14-sha256, diffie-hellman-group16-sha-512, diffie-hellman-group18-sha512, ecdh-sha2-nistp384 and ecdh-sha2-nistp521 and confirmed the TOE successfully connected using each algorithm.

FCS_SSHC_EXT.1.8

265 The evaluator needs to perform testing that rekeying is performed according to the description in the TSS. The evaluator shall test both, the time-based threshold and the traffic-based threshold.

266 For testing of the time-based threshold the evaluator shall use the TOE to connect to an SSH server and keep the session open until the threshold is reached. The evaluator shall verify that the SSH session has been active longer than the threshold value and shall verify that the TOE initiated a rekey (the method of verification shall be reported by the evaluator).

267 Testing does not necessarily have to be performed with the threshold configured at the maximum allowed value of one hour of session time but the value used for testing shall not exceed one hour. The evaluator needs to ensure that the rekeying has been initiated by the TOE and not by the SSH server the TOE is connected to.

High-Level Test Description
Using a custom SSH server, use the TOE client to connect to the server and trickle data over the channel to avoid disconnection due to idle timeout. Ensure that the TOE rekeys before 1 hour has elapsed. Ensure that the TOE is responsible for sending the rekey initiation.
Finding: Pass. The evaluator established an SSH connection between the TOE and SSH server and observed that when the 1 hour threshold was reached the TOE initiated a rekey.

268 For testing of the traffic-based threshold the evaluator shall use the TOE to connect to an SSH server and shall transmit data to and/or receive data from the TOE within the active SSH session until the threshold for data protected by either encryption key is reached. It is acceptable if the rekey occurs before the threshold is reached (e.g. because the traffic is counted according to one of the alternatives given in the Application Note for FCS_SSHC_EXT.1.8).

269 The evaluator shall verify that more data has been transmitted within the SSH session than the threshold allows and shall verify that the TOE initiated a rekey (the method of verification shall be reported by the evaluator).

270 Testing does not necessarily have to be performed with the threshold configured at the maximum allowed value of one gigabyte of transferred traffic but the value used for testing shall not exceed one gigabyte. The evaluator needs to ensure that the rekeying has been initiated by the TOE and not by the SSH server the TOE is connected to.

High-Level Test Description
Using a custom SSH server, permit the TOE client to connect to the server. The server will send large amounts of data over the channel back to the client. 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.
Finding: Pass. The evaluator established an SSH connection between the TOE and SSH server and sent continuous data through the channel. The evaluator observed that before the 1 GB threshold was reached the TOE initiated a rekey.

271 If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, the evaluator needs to verify that the threshold(s) can be configured as described in the guidance documentation and the evaluator needs to test that modification of the thresholds is restricted to Security Administrators (as required by FMT_MOF.1/Functions).

Test Not Applicable	These test limits are not configurable.

272 In cases where data transfer threshold could not be reached due to hardware limitations it is acceptable to omit testing of this (SSH rekeying based on data transfer threshold) threshold if both the following conditions are met:

a) An argument is present in the TSS section describing this hardware-based limitation and

b) All hardware components that are the basis of such argument are definitively identified in the ST. For example, if specific Ethernet Controller or WiFi radio chip is the root cause of such limitation, these chips must be identified

Test Not Applicable	The TOE does not have hardware limitations.

FCS_SSHC_EXT.1.9

273 Test 1: The evaluator shall delete all entries in the TOE's list of recognized SSH server host keys and, if selected, all entries in the TOE's list of trusted certification authorities. The evaluator shall initiate a connection from the TOE to an SSH server. The evaluator shall ensure that the TOE either rejects the connection or displays the SSH server's public key (either the key bytes themselves or a hash of the key using any allowed hash algorithm) and prompts the user to accept or deny the key before continuing the connection.

High-Level Test Description
Clear the known host key database. Using the TOE SSH client, connect to an SSH server and show that the TOE either warns the administrator that the host is unknown or that it rejects the connection attempt until after the host key has been manually added.
Finding: Pass. The evaluator cleared the TOE's known host key database and attempted a connection to the SSH server. The evaluator confirmed that the TOE prompted to confirm the SSH server's host key before continuing with the connection.

274 Test 2: The evaluator shall add an entry associating a host name with a public key into the TOE's local database. The evaluator shall replace, on the corresponding SSH server, the server's host key with a different host key. If 'password-based' is selected for the TOE in FCS_SSHC_EXT.1.2, the evaluator shall initiate a connection from the TOE to the SSH server using password-based authentication, shall ensure that the TOE rejects the connection, and shall ensure that the password was not transmitted to the SSH server (for example, by instrumenting the SSH server with a debugging capability to output received passwords). If 'password-based' is not selected for the TOE in FCS_SSHC_EXT.1.2, the evaluator shall initiate a connection from the TOE to the SSH server using public key-based authentication, and shall ensure that the TOE rejects the connection.

High-Level Test Description
Add a host key to the known hosts database either explicitly or implicitly depending on the mechanism for inserting the key. Generate a different host key for the non-TOE SSH server. Using the TOE SSH client, connect to the SSH server that advertises the wrong host key and show that the TOE rejects the connection. Verify the public key authentication fails.
Finding: Pass. The evaluator replaced the SSH server's host key with one not known by the TOE and attempted an SSH connection from the TOE to the SSH server. The evaluator confirmed that the TOE rejected the connection since the host key verification failed.

5.1.3 FCS_SSHS_EXT.1 SSH Server

5.1.3.1 TSS

FCS_SSHS_EXT.1.2

275 **[NIAP TD0631]** The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication and that this list is consistent with signature verification algorithms selected in FCS_COP.1/SigGen (e.g., accepting EC keys requires corresponding Elliptic Curve Digital Signature algorithm claims).

Findings:	[ST] Section 6.2.11 states that the TOE supports public key and password-based authentication. The public key algorithms are: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521 which is consistent with FCS_COP.1/SigGen.

276 **[NIAP TD0631]** The evaluator shall confirm that the TSS includes the description of how the TOE establishes a user identity when an SSH client presents a public key or X.509v3 certificate. For example, the TOE could verify that the SSH client's presented public key matches one that is stored within the SSH server's authorized_keys file.

Findings:	[ST] Section 6.2.11 states that the TOE establishes user identity by referencing the authorized keys file when presented with a public key authentication attempt.

277 **[NIAP TD0631]** If password-based authentication method has been selected in the FCS_SSHS_EXT.1.2, then the evaluator shall confirm its role in the authentication process is described in the TSS.

Findings:	[ST] Section 6.2.11 confirms that the TOE supports password-based authentication as an SSH server.

FCS_SSHS_EXT.1.3

278 The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled.

Findings:	[ST] Section 6.2.11 states that the TOE automatically drops packets greater than 256KB.

FCS_SSHS_EXT.1.4

279 The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

Findings:	[ST] Section 6.2.11 states that the TOE uses AES-CTR-128 and AES-CTR-256 for SSH encryption which is consistent with those claimed in FCS_SSHS_EXT.1.4.

FCS_SSHS_EXT.1.5

280 **[NIAP TD0631]** The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the SSH server's host public key algorithms supported are specified and that they are identical to those listed for this component.

Findings:	[ST] Section 6.2.11 states that the TOE supports public key authentication using ssh-rsa, rsa-sha2-256, rsa-sha2-512, and ecdsa-sha2-nistp256. This is consistent with the claims in FCS_SSHS_EXT.1.5.

FCS_SSHS_EXT.1.6

281 The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component.

Findings:	[ST] Section 6.2.11 states that the TOE provides data integrity for SSH connections via HMAC-SHA2-256 and HMAC-SHA2-512 which is consistent with FCS_SSHS_EXT.1.6.

FCS_SSHS_EXT.1.7

282 The evaluator shall check the TSS to ensure that it lists the supported key exchange algorithms, and that that list corresponds to the list in this component.

Findings:	[ST] Section 6.2.11 states that the TOE supports diffie-hellman-group14-sha1, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 which is consistent with FCS_SSHS_EXT.1.7.

FCS_SSHS_EXT.1.8

283 The evaluator shall check that the TSS specifies the following:

1. Both thresholds are checked by the TOE.

2. Rekeying is performed upon reaching the threshold that is hit first.

Findings:	[ST] Section 6.2.11 states that the TOE will re-key SSH connections after 1 hour or after an aggregate of 1 gig of data has been exchanged (whichever occurs first).

5.1.3.2 Guidance Documentation

FCS_SSHS_EXT.1.4

284 The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Findings:	[AGD] Section 3.2 states that no additional configuration is needed for the TOE SSH server connection.

FCS_SSHS_EXT.1.5

285 The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Findings:	[AGD] Section 3.2 states that no additional configuration is needed for the TOE SSH server connection.

FCS_SSHS_EXT.1.6

286 The evaluator shall also check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the “none” MAC algorithm is not allowed).

Findings:	[AGD] Section 3.2 states that no additional configuration is needed for the TOE SSH server connection.

FCS_SSHS_EXT.1.7

287 The evaluator shall also check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections with the TOE.

Findings:	[AGD] Section 3.2 states that no additional configuration is needed for the TOE SSH server connection.

FCS_SSHS_EXT.1.8

288 If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, then the evaluator shall check that the guidance documentation describes how to configure those thresholds. Either the allowed values are specified in the guidance documentation and must not exceed the limits specified in the SFR (one hour of session time, one gigabyte of transmitted traffic) or the TOE must not accept values beyond the limits specified in the SFR. The evaluator shall check that the guidance documentation describes that the TOE reacts to the first threshold reached.

Findings:	[AGD] Section 3.2 states that no additional configuration is needed for the TOE SSH server connection.

5.1.3.3 Tests

FCS_SSHS_EXT.1.2

289 **[NIAP TD0631]** Test objective: The purpose of these tests is to verify server supports each claimed client authentication method.

290 Test 1: For each supported client public-key authentication algorithm, the evaluator shall configure a remote client to present a public key corresponding to that authentication method (e.g., 2048-bit RSA key when using ssh-rsa public key). The evaluator shall establish sufficient separate SSH connections with an appropriately configured remote non-TOE SSH client to demonstrate the use of all applicable public key algorithms. It is sufficient to observe the successful completion of the SSH Authentication Protocol to satisfy the intent of this test.

High-Level Test Description
Attempt to authenticate the TOE to a SSH server using each of the claimed public-key authentication methods.
Finding: Pass. The evaluator attempted separate connections to the TOE via SSH using ssh-rsa(2048 bits), ssh-rsa(3072 bits), ssh-rsa(4096 bits), rsa-sha2-256 (2048 bits), rsa-sha2-256(3072 bits), rsa-sha2-256(4096 bits), rsa-sha2-512 (2048 bits), rsa-sha2-512 (3072 bits), rsa-sha2-512 (4096 bits), ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521 public keys and confirmed the communication was successful.

291 **[NIAP TD0631]** Test 2: The evaluator shall choose one client public key authentication algorithm supported by the TOE. The evaluator shall generate a new client key pair for that supported algorithm without configuring the TOE to recognize the associated public key for authentication. The evaluator shall use an SSH client to attempt to connect to the TOE with the new key pair and demonstrate that authentication fails.

High-Level Test Description
Add a user public key to be trusted TOE. Initiate a public key authentication attempt to the TOE from an SSH client and verify that the attempt succeeds. Generate a different user keypair for the non-TOE SSH client. Using the SSH client, connect to the TOE using the newly created keypair and show that the TOE rejects the connection.
Finding: Pass. The evaluator attempted an SSH connection to the TOE using ssh-rsa however the public key used was not known by the TOE. The evaluator confirmed that the TOE rejected the connection.

292 **[NIAP TD0631]** Test 3: [Conditional] If password-based authentication method has been selected in the FCS_SSHS_EXT.1.2, the evaluator shall configure the TOE to accept password-based authentication and demonstrate that user authentication succeeds when the correct password is provided by the connecting SSH client.

High-Level Test Description
This test was conducted as part of FIA_UIA_EXT.1.
Finding: Pass. FIA_UIA_EXT.1 Test 1 shows that the user can be authenticated with a good password.

293 **[NIAP TD0631]** Test 4: [Conditional] If password-based authentication method has been selected in the FCS_SSHS_EXT.1.2, the evaluator shall configure the TOE to accept password-based authentication and demonstrate that user authentication fails when the incorrect password is provided by the connecting SSH client.

High-Level Test Description
This test was conducted as part of FIA_UIA_EXT.1.
Finding: Pass. FIA_UIA_EXT.1 Test 1 shows that the user is not authenticated with a bad password.

FCS_SSHS_EXT.1.3

294 The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

High-Level Test Description
Using a custom tool, transmit a packet larger than the expected TOE buffer size and show that the TOE rejects the packet in some way.

High-Level Test Description
Finding: Pass. The evaluator established an SSH connection to the TOE then sent a packet larger than 256 kilobytes to the TOE and confirmed that the TOE rejected the packet and closed the connection upon receipt of the packet.

FCS_SSHS_EXT.1.4

295 The evaluator must ensure that only claimed ciphers and cryptographic primitives are used to establish a SSH connection. To verify this, the evaluator shall start session establishment for a SSH connection from a remote client (referred to as 'remote endpoint' below). 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 all the ciphers defined in the TSS for the TOE for SSH sessions, but no additional ones compared to the definition in the TSS. The evaluator shall perform one successful negotiation of an SSH session to verify that the TOE behaves as expected. It is sufficient to observe the successful negotiation of the session to satisfy the intent of the test. If the evaluator detects that not all ciphers defined in the TSS for SSH are supported by the TOE and/or the TOE supports one or more additional ciphers not defined in the TSS for SSH, the test shall be regarded as failed.

High-Level Test Description
Using an SSH client, connect to the TOE server and capture the TOE server's advertised supported cipher algorithms. Verify that the advertised set matches the claimed set. Forcibly use an SSH client to connect using only one of those ciphers and show that the connection is successful.
Finding: Pass. The evaluator established an SSH connection to the TOE and verified that the TOE supports aes128-ctr and aes256-ctr as specified by [ST].

FCS_SSHS_EXT.1.5

296 **[NIAP TD0631]** Test objective: This test case is meant to validate that the TOE server will support host public keys of the claimed algorithm types.

297 Test 1: The evaluator shall configure (only if required by the TOE) the TOE to use each of the claimed host public key algorithms. The evaluator will then use an SSH client to confirm that the client can authenticate the TOE server public key using the claimed algorithm. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

298 Has effectively been moved to FCS_SSHS_EXT.1.2.

High-Level Test Description
Using an SSH client, connect to the TOE server and capture the TOE server's host key algorithms. Verify that the client successfully connects using each claimed host key algorithm.
Finding: Pass. The evaluator successfully connected to the TOE over SSH using each of the claimed host public key algorithms.

299 **[NIAP TD0631]** Test objective: This negative test case is meant to validate that the TOE server does not support host public key algorithms that are not claimed.

300 Test 2: The evaluator shall configure a non-TOE SSH client to only allow it to authenticate an SSH server host public key algorithm that is not included in the ST

selection. The evaluator shall attempt to establish an SSH connection from the non-TOE SSH client to the TOE SSH server and observe that the connection is rejected.

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.
Finding: Pass. The evaluator attempted an SSH connection to the TOE requesting the ssh-ed25519 host key algorithm and confirmed that the TOE rejected the connection.

FCS_SSHS_EXT.1.6

301 Test 1: (conditional, if an HMAC or AEAD_AES_*_GCM algorithm is selected in the ST) The evaluator shall establish an SSH connection using each of the algorithms, except “implicit”, specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

302 Note: To ensure the observed algorithm is used, the evaluator shall ensure a non-aes*-gcm@openssh.com encryption algorithm is negotiated while performing this test.

High-Level Test Description
Using an SSH client, forcibly negotiate only the claimed integrity algorithms and show that they are accepted to form a successful connection.
Finding: Pass. The evaluator attempted separate SSH connections to the TOE using hmac-sha2-256 and hmac-sha2-512 and confirmed the TOE successfully connected using each algorithm.

303 Test 2: (conditional, if an HMAC or AEAD_AES_*_GCM algorithm is selected in the ST) The evaluator shall configure an SSH client to only allow a MAC algorithm that is not included in the ST selection. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.

304 Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*-gcm@openssh.com encryption algorithm is negotiated while performing this test.

High-Level Test Description
Using an SSH client, forcibly negotiate an integrity algorithm which is not claimed by the TOE and show that it results in a failed connection.
Finding: Pass. The evaluator attempted an SSH connection to the TOE using hmac-md5 and confirmed that the TOE rejected the connection.

FCS_SSHS_EXT.1.7

305 Test 1: The evaluator shall configure an SSH client to only allow the diffie-hellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.

High-Level Test Description
Using an SSH client, forcibly negotiate the diffie-hellman-group1-sha1 key exchange algorithm which is not supported by the TOE and show that it results in a failed connection.
Finding: Pass. The evaluator attempted an SSH connection to the TOE using diffie-hellman-group1-sha1 and confirmed that the TOE rejected the connection.

306 Test 2: For each allowed key exchange method, the evaluator shall configure an SSH client to only allow that method for key exchange, attempt to connect from the client to the TOE, and observe that the attempt succeeds.

High-Level Test Description
Using an SSH client, forcibly negotiate each of the claimed key exchange algorithms in turn and show that it results in a successful connection.
Finding: Pass. The evaluator attempted separate SSH connections to the TOE using diffie-hellman-group14-sha1, ecdh-sha2-nistp256, diffie-hellman-group14-sha256, diffie-hellman-group16-sha-512, diffie-hellman-group18-sha512, ecdh-sha2-nistp384 and ecdh-sha2-nistp521 and confirmed the TOE successfully connected using each algorithm.

FCS_SSHS_EXT.1.8

307 The evaluator needs to perform testing that rekeying is performed according to the description in the TSS. The evaluator shall test both, the time-based threshold and the traffic-based threshold.

308 For testing of the time-based threshold the evaluator shall use an SSH client to connect to the TOE and keep the session open until the threshold is reached. The evaluator shall verify that the SSH session has been active longer than the threshold value and shall verify that the TOE initiated a rekey (the method of verification shall be reported by the evaluator).

309 Testing does not necessarily have to be performed with the threshold configured at the maximum allowed value of one hour of session time but the value used for testing shall not exceed one hour. The evaluator needs to ensure that the rekeying has been initiated by the TOE and not by the SSH client that is connected to the TOE.

High-Level Test Description
Using a custom SSH client, connect to the TOE and trickle data over the channel to avoid disconnection due to idle timeout. Ensure that the TOE rekeys before 1 hour has elapsed. Ensure that the TOE is responsible for sending the rekey initiation.
Finding: Pass. The evaluator established an SSH connection to the TOE and observed that when the 1 hour threshold was reached the TOE initiated a rekey.

310 For testing of the traffic-based threshold the evaluator shall use the TOE to connect to an SSH client and shall transmit data to and/or receive data from the TOE within the active SSH session until the threshold for data protected by either encryption key is reached. It is acceptable if the rekey occurs before the threshold is reached (e.g. because the traffic is counted according to one of the alternatives given in the Application Note for FCS_SSHS_EXT.1.8).

311 The evaluator shall verify that more data has been transmitted within the SSH session than the threshold allows and shall verify that the TOE initiated a rekey (the method of verification shall be reported by the evaluator).

312 The evaluator shall verify that more data has been transmitted within the SSH session than the threshold allows and a corresponding audit event has been generated by the TOE.

313 Testing does not necessarily have to be performed with the threshold configured at the maximum allowed value of one gigabyte of transferred traffic but the value used for testing shall not exceed one gigabyte. The evaluator needs to ensure that the rekeying has been initiated by the TOE and not by the SSH client that is connected to the TOE.

High-Level Test Description
Using a custom SSH client, connect to the TOE 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.
Finding: Pass. The evaluator established an SSH connection to the TOE and sent continuous data through the channel. The evaluator observed that before the 1 GB threshold was reached the TOE initiated a rekey.

314 If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, the evaluator needs to verify that the threshold(s) can be configured as described in the guidance documentation and the evaluator needs to test that modification of the thresholds is restricted to Security Administrators (as required by FMT_MOF.1/Functions).

Test Not Applicable	These limits are not configurable for this TOE.

315 In cases where data transfer threshold could not be reached due to hardware limitations it is acceptable to omit testing of this (SSH rekeying based on data transfer threshold) threshold if both the following conditions are met:

a. An argument is present in the TSS section describing this hardware-based limitation and

b. All hardware components that are the basis of such argument are definitively identified in the ST. For example, if specific Ethernet Controller or WiFi radio chip is the root cause of such limitation, these chips must be identified.

Test Not Applicable	The TOE does not have hardware limitations.

5.2 Security management (FMT)

5.2.1 FMT_MOF.1/Functions Management of security functions behaviour

5.2.1.1 TSS

316 For distributed TOEs see [SD] chapter 2.4.1.1

Findings:	The TOE is not a distributed TOE.

317 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:	[ST] Section 6.4.1 states that the TOE defines a single role of Security Administrator. The Security Administrator is able to start and stop the trusted path / trusted channels via the CLI. The Security Administrator is also able to modify the behaviour of audit data to an external IT entity. This section also states that the administrator can modify the behaviour of the audit data transmission by enabling and disabling the syslog service, configuring the reference identifier of the remote server and configuring the public key used to authenticate to the remote server.

5.2.1.2 Guidance Documentation

318 For distributed TOEs see [SD] chapter 2.4.1.2.

Findings:	The TOE is not a distributed TOE.

319 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:	[AGD] Section 3.6 “Audit logging” contains instructions on transmitting audit data to an external IT entity.

5.2.1.3 Tests

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

High-Level Test Description
The evaluator shall attempt to modify the security related parameters without prior authentication as a Security Administrator and show that the attempt is unsuccessful.
Finding: Pass. The evaluator confirmed that the TOE requires the Security Administrator to be authenticated before the transmission of audit data parameters can be modified.

321 Test 2 (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): 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.

High-Level Test Description
Using the privileged ‘admin’ user modify the security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity. Show that the TOE successfully connects to the newly configured external audit server.
Finding: Pass. The evaluator confirmed that the TOE enforces transmission of audit data parameters after they are changed by the Security Administrator.

322 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.

323 Test 1 (if 'handling of audit data' is selected from the second selection together with 'modify the behaviour of' in the first selection): 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.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace.

Test Not Applicable The [ST] does not claim this functionality.

324 Test 2 (if 'handling of audit data' is selected from the second selection together with 'modify the behaviour of' in the first selection): 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.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace.

325 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.

Test Not Applicable The [ST] does not claim this functionality.

326 Test 1 (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): 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.

327 Test 2 (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): 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.

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

Test Not Applicable	The [ST] does not claim this functionality.

329 Test 3 (if in the first selection 'determine the behaviour of' has been chosen together with for any of the options in the second selection): 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.

330 Test 4 (if in the first selection 'determine the behaviour of' has been chosen together with for any of the options in the second selection): 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 administrator authentication shall be successful.

Test Not Applicable	The [ST] does not claim this functionality.

5.2.2 FMT_MOF.1/Services Management of Security Functions Behaviour

5.2.2.1 TSS

331 For distributed TOEs see [SD] chapter 2.4.1.1.

Findings:	The TOE is not a distributed TOE.

332 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 how that operation is performed.

Findings:	[ST] Section 6.4.3 states that the Security Administrators can start and stop the syslog and ntp services via the CLI.

5.2.2.2 Guidance Documentation

333 For distributed TOEs see [SD] chapter 2.4.1.2.

Findings:	The TOE is not a distributed TOE.

334 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 how that operation is performed.

Findings:	[AGD] Section 3.5 states that the NTP service is stopped when there are no NTP servers configured. Section 3.6 also includes instructions for enabling and disabling the remote syslog service.

5.2.2.3 Tests

335 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.

336 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 add an SSH key to the TOE without authenticating as an administrator and show the attempt is unsuccessful.
Finding: Pass. The evaluator confirmed that the TOE requires the Security Administrator to be authenticated before an SSH key can be added.

5.2.3 FMT_MTD.1/CryptoKeys Management of TSF Data

5.2.3.1 TSS

337 For distributed TOEs see [SD] chapter 2.4.1.1.

Findings:	The TOE is not a distributed TOE.

338 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 how that how those operations are performed.

Findings:	[ST] Section 6.4.6 states that the TOE restricts the ability to manage SSH keys to Security Administrators.

5.2.3.2 Guidance Documentation

339 For distributed TOEs see [SD] chapter 2.4.1.2.

Findings:	The TOE is not a distributed TOE.

340 For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation 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 how that how those operations are performed.

Findings:	[AGD] section 3.3 contains instructions for generating, adding and removing keys.

5.2.3.3 Tests

341 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 generate a new SSH key to the TOE without prior authentication and show the attempt is not successful.
Finding: Pass. The evaluator confirmed that the TOE requires the Security Administrator to be authenticated before a new SSH key can be generated.

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

High-Level Test Description
Attempt to generate a new SSH key to the TOE with prior authentication and show the attempt is successful.
Finding: Pass. The evaluator confirmed that the TOE allows the Security Administrator to generate a new SSH key after authentication.

6 Evaluation Activities for Security Assurance Requirements

6.1 ASE: Security Target

343 When evaluating a Security Target, the evaluator performs the work units as presented in the CEM. In addition, the evaluator ensures the content of the TSS in the ST satisfies the EAs specified in Section 2 (Evaluation Activities for SFRs).

Findings:	All work units presented in the CEM are addressed in the [ETR]. See the above sections for coverage of the TSS EAs.

344 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:	N/A since the TOE is not a distributed TOE.

6.2 ADV: Development

345 The design information about the TOE is contained in the guidance documentation available to the end user as well as the TSS portion of the ST, and any required supplementary information required by this cPP that is not to be made public.

346 The functional specification describes the TOE Security Functions Interfaces (TSFIs). It is not necessary to have a formal or complete specification of these interfaces.

347 No additional "functional specification" documentation is necessary to satisfy the Evaluation Activities specified in [SD].

348 The Evaluation Activities in [SD] are associated with the applicable SFRs; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

349 5.2.1.1 Evaluation Activity: 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.

Findings:	From section 7.2.1 of the NDcPP : “For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation.” The [ST] and the AGD comprise the functional specification. If the test in [SD] cannot be completed because the [ST] or the AGD are incomplete, then the functional specification is not complete and observations are required. During the evaluator’s use of the product and its interfaces (the SSH CLI and local console port), there were no areas that were deficient.

350 5.2.1.2 Evaluation Activity: 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.

Findings:	See comments in the previous work unit.

351 5.2.1.3 Evaluation Activity: The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

Findings:	See comments in the previous work unit.

6.2.1 AGD: Guidance

352 The design information about the TOE is contained in the guidance documentation available to the end user as well as the TSS portion of the ST, and any required supplementary information required by this cPP that is not to be made public.

353 5.3.1.1 Evaluation Activity: 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:	The [AGD] in conjunction with [ADMIN] documentation is available on the NIAP PCL for administrators to download and use to configure the TOE into the evaluated configuration.

354 5.3.1.2 Evaluation Activity: 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:	There is only one operational environment claimed in the [ST]. All TOE platforms claimed in [ST] are covered by the operational guidance. The [ST] in conjunction with the [AGD] and [ADMIN] appropriately cover the components needed in the operational environment and the configuration needed to communicate with them.

355 5.3.1.3 Evaluation Activity: The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

Findings:	The [AGD] section 3.3. covers the configuration of the cryptographic engine. This section states that FIPS mode is enabled by default and no further configuration is needed to achieve the evaluated configuration of the cryptographic module.

356 5.3.1.4 Evaluation Activity: 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:	The [AGD] document section 1.3.3 covers configuration of the evaluated functionality where additional configuration might be required. Additionally, [AGD] section 3.2 covers the administrative interfaces included in the evaluation.

5.3.1.5 Evaluation Activity

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

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

[NIAP TD0536] b) The documentation must describe the process for verifying updates to the TOE for each method selected for FPT_TUD_EXT.1.3 in the Security Target. The evaluator shall verify that this process includes the following steps:

5) 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).

6) Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes instructions that describe at least one method of validating the hash/digital signature.

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:	See work unit [PP] 5.3.1.3 for configuration of the cryptographic engine. [AGD] Section 2.4 states that the TOE software is verified via digital signature by the TOE and the upgrade packages are delivered directly by the vendor. See work unit [PP] 5.3.1.4 for details as to what was covered by the EAs.

358 5.3.2.1 Evaluation Activity: The evaluator shall examine the Preparative procedures to ensure they include a description of how the 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).

Findings:	This activity is covered by the [ETR] work unit AGD_OPE.1-6.

359 5.3.2.2 Evaluation Activity: 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:	There is only one operational environment claimed in the [ST]. All TOE platforms claimed in [ST] are covered by the operational guidance. The operational guidance is evidenced by [AGD] and [ADMIN].

360 5.3.2.3 Evaluation Activity: The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.

Findings:	[ST] Section 2.2.2 identifies a single operational environment for the TOE. The guidance provided by [AGD] and [ADMIN] describe how the TOE is installed and configured in this operational environment.

361 5.3.2.4 Evaluation Activity: The evaluator shall examine the preparative procedures to ensure they include instructions to manage the security of the TSF as a product and as a component of the larger operational environment.

Findings:	The [AGD] and [ADMIN] guidance documentation provides information on managing the security of the TOE as an individual product and as a component of the operational environment.

362 5.3.2.5 Evaluation Activity: In addition the evaluator shall ensure that the following requirements are also met.

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 instructions shall be provided for how these can be changed.

Findings:	[AGD] Section 3.2 discusses the administrative interfaces and provide instructions for configuring the parameters associated with administrative access. [AGD] Section 3.4 also discusses default passwords and instructs the admin to follow [ADMIN] section “Change Password” to set a new password.

6.3 ALC: Life-cycle Support

6.3.1 Labelling of the TOE (ALC_CMC.1)

363 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:	These work units were completed in the [ETR].

6.3.2 TOE CM coverage (ALC_CMS.1)

364 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:	These work units were completed in the [ETR].

6.4 ATE: Tests

6.4.1 Independent Testing – Conformance (ATE_IND.1)

365 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.

366 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 Sections 2, 3 and 4.

367 The evaluator should consult Appendix B when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

368 Note that additional Evaluation Activities relating to evaluator testing in the case of a distributed TOE are defined in section A.9.3.1.

Findings:	The evaluator provided a Detailed Test Report [DTR] to satisfy this requirement. The [DTR] includes details of the components in the test environment, the TOE, test tools, test cases, and expected and actual results.

6.5 Vulnerability Assessment

6.5.1 Vulnerability Survey (AVA_VAN.1)

369 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 follows 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

370 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.

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

372 5.6.1.1 Evaluation Activity (Documentation): In addition to the activities specified by the CEM in accordance with Table 2, the evaluator shall perform the following activities.

373 *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.*

[NIAP TD0547]

374 The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components should identify at a minimum the processors 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 libraries, (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:	The evaluator collected this information from the developer which was used to feed into the Type 1 Flaw Hypotheses search (below).

375 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, 6.3.3]

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.

376 5.6.1.2 Evaluation Activity: The evaluator formulates hypotheses in accordance with process defined in Appendix A. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3.

Findings: 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:

NIST National Vulnerabilities Database (can be used to access CVE and US-CERT databases identified below): https://web.nvd.nist.gov/view/vuln/search

Common Vulnerabilities and Exposures:
https://cve.mitre.org/cve/search_cve_list.html

US-CERT: http://www.kb.cert.org/vuls/html/search

Tenable Network Security: https://www.tenable.com/cve

Tipping Point Zero Day Initiative: https://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 last conducted on November 9, 2022 and included the following search terms[3]:

CAE MPIC

CAE

i.MX6

ARM Cortex-A9

iptables

Linux kernel

openssh

openssl

ntpd

The evaluation team determined that there are no residual vulnerabilities based on these searches.

[3] Note the precise software versions were used and provided in the proprietary vulnerability analysis document.

The type-2 hypotheses – iTC-Sourced test identified in not applicable to the TOE since it does not include a TLS Server. The evaluation team developed Type 3 flaw hypotheses in accordance with Sections A.1.3, A.1.4, 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 Sections A.1.3, A.1.4, and A.2, and no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.