Email Security Appliance

Assurance Activity Report

Version 1.0

September 2024

Document prepared by

Lightship Security

www.lightshipsec.com

Document History

Version	Date	Author	Reviewer	Description
0.1	2024-08-06	K. Steiner	C. Cantlon	Initial draft
0.2	2024-09-06	K. Steiner		ECR Update
1.0	2024-09-12	K. Steiner		Release for PCL

Table of Contents

1 Introduction

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

1.1 Evaluation Identifiers

Table 1: Evaluation Identifiers

Scheme	NIAP Common Criteria Evaluation and Validation Scheme
Evaluation Facility	Lightship Security
Developer/Sponsor	Cisco Systems, Inc.
TOE	Email Security Appliance
Security Target	Cisco Email Security Appliance Security Target, Version 1.0, 12 September 2024
Protection Profile	collaborative Protection Profile for Network Devices, Version 2.2e, 23-March-2020 [NDcPP]

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, December-2019, Version 2.2 [ND-SD]

Table 3: Interpretations

NDcPP v2.2e Technical Decisions	Applicable
TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes
TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No—Referenced SFR (FCS_NTP_EXT.1) is not being claimed.
TD0536: NIT Technical Decision for Update Verification Inconsistency	Yes

NDcPP v2.2e Technical Decisions	Applicable
TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Yes
TD0546: NIT Technical Decision for DTLS – clarification of Application Note 63	No – Referenced SFRs are not being claimed.
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	Yes
TD0556: NIT Technical Decision for RFC 5077 question	Yes
TD0563: NiT Technical Decision for Clarification of audit date information	Yes
TD0564: NiT Technical Decision for Vulnerability Analysis Search Criteria	Yes
TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	Yes
TD0570: NiT Technical Decision for Clarification about FIA_AFL.1	Yes
TD0571: NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes
TD0572: NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes
TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes
TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes
TD0591: NIT Technical Decision for Virtual TOEs and hypervisors	No – virtual TOE meets the criteria for Case 1 (described in 2.2e section 1.2) and thus the TD doesn’t apply
TD0592: NIT Technical Decision for Local Storage of Audit Records	Yes
TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes
TD0632: NIT Technical Decision for Consistency with Time Data for vNDs	Yes

NDcPP v2.2e Technical Decisions	Applicable
TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters	Yes
TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH	Yes
TD0638: NIT Technical Decision for Key Pair Generation for Authentication	Yes
TD0639: NIT Technical Decision for Clarification for NTP MAC Keys	No – Referenced SFR is not being claimed.
TD0670: NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	No – Referenced SFR is not being claimed.
TD0738: NIT Technical Decision for Link to Allowed-With List	Yes
TD0790: NIT Technical Decision: Clarification Required for testing IPv6	No – Referenced SFRs are not being claimed.
TD0792: NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes
TD0800: Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	No – Referenced SFR is not being claimed.

1.3 Reference Documents

Table 4: List of Reference Documents

Ref	Document
[ST]	Cisco Email Security Appliance Security Target, Version 1.0, 12 September 2024
[AGD]	Cisco Email Security Appliance running AsyncOS 15.5 Common Criteria Operational User Guidance And Preparative Procedures, Version v1.0, September 12, 2024
[DTR]	Cisco Email Security Appliance NDcPPv2.2E Detailed Test Report, Version 0.4, September 2024 Cisco Email Security Appliance NDcPPv2.2E Test Results, Version 0.4, September 2024
[ETR]	Cisco Email Security Appliance Evaluation Technical Report, Version 0.4, September 2024
[AVA]	Cisco Email Security Appliance with AsyncOS 15.5 Vulnerability Assessment, Version 0.2, September 2024

2 TOE Overview

3 The TOE, which consists of the Cisco Email Security Appliance, is a network device. ESA is an appliance that provides comprehensive email protection services for a company's email system. It is an email protection product that monitors Simple Mail Transfer Protocol (SMTP) network traffic, analyzes the monitored network traffic using various techniques, and reacts to identified threats associated with email messages (such as spam and inappropriate or malicious content). The TOE includes the hardware models as defined in [ST] Table 3 in section 1.1.

2.1 TOE Models

4 Refer to [ST] section 1.5, Table 5.

2.1.1 Test Platform Equivalency

5 The evaluator performed full end-to-end testing on the Cisco C395 and Cisco C100v models. Detailed equivalency rationale is provided in [DTR].

3 Evaluation Activities for Mandatory SFRs

3.1 Security Audit (FAU)

3.1.1 FAU_GEN.1 Audit data generation

3.1.1.1 TSS

6 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
PASS
In section 6.1 of the [ST] under FAU_GEN.1, the TSS states that the TOE logs a reference to any associated keys.

7 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
Not Applicable: The TOE is not a distributed TOE.

3.1.1.2 Guidance Documentation

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

Findings
PASS
[AGD] section 5 contains Table 7 which provides an example of each auditable event required by FAU_GEN.1.

9 The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including

enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

Findings
PASS
[AGD] sections 3 and 4 provide instructions for using the TOE according to the requirements specified in the [ST], including commands to run and compliant parameters. [AGD] sections 1.5 and 1.6 provide general guidance regarding the scope of the evaluated functionality.

3.1.1.3 Tests

10 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.

Findings
PASS
The evaluator performed the testing in conjunction with the testing of the security mechanisms directly. The evaluator confirmed that the TOE correctly generates audit records for the events listed in the table of audit events and administrative actions.

11 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.

12 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

13 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

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

15 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

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

Findings
PASS
The [ST] in section 6.1 under FAU_STG_EXT.1 the TSS indicates that the TOE is configured by the Administrator to send logs to a remote server using SCP over SSHv2.

17 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
PASS
In section 6.1 in the [ST], under FAU_STG_EXT.1, the TSS states that by default the TOE maintains 10 log files of no more than 10MB for each log subscription. The TOE also allows the Authorized Administrator to configure each log subscription to a number of logs up to a configurable size limit. If the space available for storing audit records is exhausted, the TOE will start to overwrite the oldest records. Only Authorized Administrators can clear the local logs, and there is no TOE interface that allows for administrators to modify the contents of the local audit records.

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

Findings
Not applicable: The TOE is not a distributed TOE.

19 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
PASS
As in the previous work units, section 6.1 of the [ST] states that if the space available for storing audit records is exhausted, the TOE will start to overwrite the oldest records.

20 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
PASS
Section 6.1 of the [ST] under FAU_STG_EXT.1 indicates that the transmission is done based on configurable time and space rules provided by the administrator.

21 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
Not Applicable: The TOE is not a distributed TOE.

22 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
Not Applicable: The TOE is not a distributed TOE.

3.1.3.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 3.4 describes the logging configuration. Logging to an external SCP server uses SSHv2 and is configured using the ‘logconfig’ command.

24 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
PASS
[AGD] Section 3.4 states that once remote logging to the SCP server is configured, the logs are periodically pushed. The time period is based on configured time intervals.

25 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
PASS
[AGD] Section 3.4 describes the log rollover settings. The ‘logconfig’ command is used to configure the maximum file size per FAU_STG_EXT.1.3.

3.1.3.3 Tests

26 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.

Findings
PASS
The evaluator confirmed the TOE sends audit logs to a remote server via encrypted SSH without administrator intervention and was successfully received by the audit server. OpenSSH 9.6p1 was used as an SCP listener as the remove audit server.

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).

Findings
PASS
The evaluator confirmed that the TOE overwrites the oldest log file when the configured storage space for audit logs is filled.

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 is not compliant with FAU_STG_EXT.2/LocSpace.

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 NIAP Policy 5

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

SFR	Cryptographic Operation	NIST Standard	CAVP Certificate (Algorithm)
FCS_CKM.1	Asymmetric Key Generation: RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3;	FIPS PUB 186-4	A4446 (RSA) A4595 (RSA)
FCS_CKM.1	Asymmetric Key Generation: ECC schemes using “NIST curves” [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4;	FIPS PUB 186-4	A4446 (ECDSA) A4595 (ECDSA)
FCS_CKM.1	Asymmetric Key Generation: FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.1	FIPS PUB 186-4	A4446 (DSA) A4595 (DSA)
FCS_CKM.2	RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1”;	N/A	CCTL Tested
FCS_CKM.2	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;	NIST SP 800-56A Revision 3	A4446 (KAS-ECC-SSC) A4595 (KAS-ECC-SSC)
FCS_CKM.2	Finite field-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key	NIST SP 800-56A Revision 2	A4446 (KAS-FFC-SSC)

SFR	Cryptographic Operation	NIST Standard	CAVP Certificate (Algorithm)
	Establishment Schemes Using Discrete Logarithm Cryptography”;		A4595 (KAS-FFC-SSC)
FCS_COP.1/ DataEncryption	encryption/decryption in accordance with a specified cryptographic algorithm AES used in [CBC, CTR, GCM] mode and cryptographic key sizes [128 bits, 256 bits]	FIPS PUB 197 NIST SP 800-38A NIST SP 800-38D	A4446 (AES-CBC, AES-CTR, AES-GCM) A4595 (AES-CBC, AES-CTR, AES-GCM)
FCS_COP.1/ SigGen	cryptographic signature services (generation and verification): RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits or greater],	FIPS PUB 186-4	A4446 (RSA) A4595 (RSA)
FCS_COP.1/ SigGen	cryptographic signature services (generation and verification): Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits]	FIPS PUB 186-4	A4446 (ECDSA) A4595 (ECDSA)
FCS_COP.1/ Hash	cryptographic hashing services in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512]	FIPS PUB 180-4	A4446 (SHA-1, SHA2-256, SHA2-384, SHA2-512) A4595 (SHA-1, SHA2-256, SHA2-384, SHA2-512)
FCS_COP.1/ KeyedHash	keyed-hash message authentication in accordance with a specified cryptographic algorithm [selection: HMAC-SHA-1, HMAC-SHA-256]	FIPS PUB 198-1	A4446 (HMAC-SHA-1, HMAC_SHA2-256) A4595 (HMAC-SHA-1, HMAC_SHA2-256)
FCS_RBG_EXT.1	random bit generation services using [CTR_DRBG (AES)]	NIST SP 800-90A Rev. 1	A4446 (Counter DRBG) A4595 (Counter DRBG)

3.2.2 FCS_CKM.1 Cryptographic Key Generation

3.2.2.1 TSS

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

Findings
PASS
This information can be found in section 6.1 of the [ST] under FCS_CKM.1 and FCS_CKM.2. Specifically, the TOE can generate 2048-bit (or greater) RSA keys for use in a Certificate Signing Request. The TOE also acts as a sender and receiver for Diffie-Hellman and EC Diffie-Hellman key establishment and is responsible for key generation (2048-bit, P-256, P-384 and P-521) for key establishment. The [ST] in section 6.1 for FCS_CKM.1 and FCS_CKM.2 provides a sub-table showing how each scheme is used. This sub-table is consistent with the claims and the testing.

3.2.2.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 3.2.1 describes how to configure the TOE in FIPS mode. Once in FIPS mode, the FIPS approved algorithms and key sizes are automatically configured.

3.2.2.3 Tests

30 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

31 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*.

32 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

33 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

34 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

35 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)

36 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.

37 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

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

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

39 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.

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

41 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.

42 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

43 for each FFC parameter set and key pair.

NIAP TD0580

FFC Schemes using "safe-prime"

NIAP TD0580

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

Findings
🟩 PASS
[ST] Table 7 specifies the CAVP certificate demonstrating the TOE correctly implements RSA, ECDSA and FFC key generation.

3.2.3 FCS_CKM.2 Cryptographic Key Establishment

3.2.3.1 TSS

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

Findings
🟩 PASS
This information can be found in section 6.1 of the [ST] under FCS_CKM.1 and FCS_CKM.2. The TOE acts as a sender and receiver for RSA, Diffie-Hellman and EC Diffie-Hellman key establishment and is responsible for key generation (2048-bit, P-256, P-384 and P-521) for key establishment.

NIAP TD0580

46 **Removed:** ~~If Diffie-Hellman group 14 is selected from FCS_CKM.2.1, the TSS shall claim the TOE meets RFC 3526 Section 3.~~

47 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
~~Diffie-Hellman (Group 14)~~ **Removed per TD0580**	~~FCS_SSHC_EXT.1~~ **Removed per TD0580**	~~Backup Server~~ **Removed per TD0580**
ECDH	FCS_IPSEC_EXT.1	Authentication Server

48 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
PASS
The [ST] in section 6.1 for FCS_CKM.1 and FCS_CKM.2 provides a sub-table showing how each scheme is used. This sub-table is consistent with the claims and the testing.

3.2.3.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 3.2.1 describes how to configure the TOE in FIPS mode. Once in FIPS mode, the FIPS approved algorithms and key sizes are automatically configured.

3.2.3.3 Tests

Key Establishment Schemes

50 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

51 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

52 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.

53 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.

54 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.

55 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.

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

Validity Test

57 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.

58 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).

59 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.

Findings

PASS
[ST] Table 7 specifies the CAVP certificate demonstrating the TOE correctly implements ECC and SP 800-56A key agreement/establishment schemes.

RSA-based key establishment schemes

60 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.

Findings
PASS
The evaluator confirmed the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 key establishment by successfully connecting to a "known-good" implementation as part of testing for FCS_TLSS_EXT.1.1. The "known-good" implementations used in these tests was OpenSSL 3.0.8.

NIAP TD0580 Removed:

~~***Diffie-Hellman Group 14***~~

61 ~~The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 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 Diffie-Hellman group 14.~~

FFC Schemes using "safe-prime" groups

62 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
PASS
The [ST] does not select FFC Schemes using "safe-prime" groups.

3.2.4 FCS_CKM.4 Cryptographic Key Destruction

3.2.4.1 TSS

63 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
PASS
Section 6.1 of the [ST] provides information on how keys and CSPs are zeroized. Additional information is provided in section 7 of the [ST] in the form of a table outlining the keys, their storage means and how they are zeroized. The types of keys and CSPs are consistent with the claims made in section 5 of the [ST].

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

Findings
PASS
The table included in section 7 of the [ST] indicates that the keys are overwritten with specific values using the cryptographic module to overwrite memory directly. For other keys, they are overwritten when a new key is generated.

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

Findings
PASS
The [ST] in section 5 claims that the TOE uses an interface provided by a part of the TSF. However, the interfaces are not exposed via the user-serviceable CLI or Web UI and are done automatically by the cryptographic module per TSS section 6.1, FCS_CKM.4.

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

Findings
PASS
N/A – Section 7 in the [ST] only lists keys stored in plaintext form.

[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.

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

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

68 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
Not Applicable: The use of "*a value that does not contain any CSP*" is not included in the ST.

3.2.4.2 Guidance Documentation

69 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.

70 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
PASS
The [ST] does not identify a configuration or circumstance that may not conform to the key destruction requirement.

3.2.4.3 Tests

71 None

[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).

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

3.2.5.1 TSS

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

Findings
PASS
FCS_COP.1/DataEncryption in [ST] section 6.1 states that AES is used in CBC, CTR, and GCM modes with size 128 and 256 bits is used for encryption and decryption.

3.2.5.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 3.2.1 describes how to configure the TOE in FIPS mode. Once in FIPS mode, the FIPS approved algorithms and key sizes are automatically configured.

3.2.5.3 Tests

AES-CBC Known Answer Tests

74 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.

75 **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.

76 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.

77 **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.

78 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.

79 **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].

80 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.

81 **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].

82 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

83 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.

84 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

85 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]
```

86 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.

87 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

88 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.

89 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.

90 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.

91 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

92 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):

93 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.

94 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.

95 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.

96 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].

97 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

98 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

99 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]
```

100 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.

101 There is no need to test the decryption engine.

Findings
PASS
[ST] Table 7 specifies the CAVP certificate demonstrating the TOE correctly implements AES.

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

3.2.6.1 TSS

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

Findings
PASS
FCS_COP.1/SigGen in [ST] section 6.1 states that the TOE uses RSA Digital Signature Algorithm with key size of 2048 and greater and ECDSA with key size 256 bits for cryptographic signature services.

3.2.6.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 3.2.1 describes how to configure the TOE in FIPS mode. Once in FIPS mode, the FIPS approved algorithms and key sizes are automatically configured.

3.2.6.3 Tests

ECDSA Algorithm Tests

ECDSA FIPS 186-4 Signature Generation Test

104 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

105 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

106 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.

107 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

108 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.

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

Findings
PASS
[ST] Table 7 specifies the CAVP certificate demonstrating the TOE correctly implements RSA and ECDSA signature generation and verification.

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

3.2.7.1 TSS

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

Findings
PASS
FCS_COP.1/Hash in [ST] section 6.1 states that the TOE uses SHA-1, SHA-256, SHA-384, and SHA-512. Hashes are used for TLS session integrity. SHA-384 is used for verification of software image integrity.

3.2.7.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 3.2.1 describes how to configure the TOE in FIPS mode. Once in FIPS mode, the FIPS approved algorithms and key sizes are automatically configured.

3.2.7.3 Tests

112 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.

113 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

114 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

115 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

116 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

117 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

118 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
PASS
[ST] Table 7 specifies the CAVP certificate demonstrating the TOE correctly implements hashing.

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

3.2.8.1 TSS

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

Findings
PASS
The [ST] in section 6.1 for the table entry FCS_COP.1/Hash, FCS_COP.1/KeyedHash indicates that the TOE uses HMAC-SHA-1 and HMAC-SHA-256. The key length is 160-bits, hash function is SHA1, block size is 512 bits and output MAC length is 160 bits for HMAC-SHA-1. The key length is 256-bits, hash function is SHA256, block size is 512 bits and output MAC length is 256 bits for HMAC-SHA-256.

3.2.8.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 3.2.1 describes how to configure the TOE in FIPS mode. Once in FIPS mode, the FIPS approved algorithms and key sizes are automatically configured.

3.2.8.3 Tests

121 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

PASS
[ST] Table 7 specifies the CAVP certificate demonstrating the TOE correctly implements HMAC algorithms.

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

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

3.2.9.1 TSS

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

Findings
PASS
According to section 6.1 of the [ST] under FCS_RBG_EXT.1, “The TOE implements a NIST-approved AES-CTR Deterministic Random Bit Generator (DRBG), as specified in SP 800-90 seeded by an entropy source that accumulates entropy from a TSF-software based noise source as described in FCS_RBG_EXT.1. This output is used directly to seed the DRBG.” Furthermore, the [ST] states “The deterministic RBG is seeded with a minimum of 256 bits of entropy, …”

3.2.9.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 3.2.1 describes how to configure the TOE in FIPS mode. Once in FIPS mode, the FIPS approved algorithms and key sizes are automatically configured.

3.2.9.3 Tests

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

126 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).

127 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.

128 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
PASS
[ST] Table 7 specifies the CAVP certificate demonstrating the TOE correctly implements Deterministic Random Bit Generation.

3.3 Identification and Authentication (FIA)

3.3.1 FIA_AFL.1 Authentication Failure Management

3.3.1.1 TSS

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

Findings
PASS
As per section 6.1 of the [ST] under FIA_AFL.1: “When the Authorized Administrator attempting to log into the administrative CLI or GUI interface reaches the administratively set maximum number of failed authentication attempts, the user will not be granted access to the administrative functionality of the TOE until an Authorized Administrator resets the user's number of failed login attempts through the administrative CLI using the *userconfig* command or GUI Edit User webpage.”

130 The evaluator shall examine the TSS to confirm that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access

is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking).

Findings
PASS
Section 6.1 of the [ST] under FIA_AFL.1 informs the reader that the default "admin" account is not subject to the lock out at the local console. This is to ensure the administrators do not get totally locked out of the TOE.

3.3.1.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 4.5.1 describes how to configure the failed attempt lockout threshold. This is settable for a range of 1-60 attempts, the default is 5 and the evaluated configuration requires it to be set to 3. This section also provides instructions to unlock a locked account.

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

Findings
PASS
[AGD] Section 4.5.1 states that the default admin account is not subject to the lockout criteria.

3.3.1.3 Tests

133 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.

Findings
PASS

The evaluator confirmed that the administrator is able to configure the account policy, and once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful.

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).

Findings

PASS

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

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.

Findings

Not Applicable: The ST does not select time period.

3.3.2 FIA_PMG_EXT.1 Password Management

3.3.2.1 TSS

134 **NIAP TD0792**

135 The evaluator shall check that the TSS lists the supported special character(s) for the composition of administrator passwords.

136 The evaluator shall check the TSS to ensure that the minimum_password_length parameter is configurable by a Security Administrator.

137 The evaluator shall check that the TSS lists the range of values supported for the minimum_password_length parameter. The listed range shall include the value of 15.

Findings

PASS

As per section 6.1 of the [ST] under FIA_PMG_EXT.1 the TOE supports the special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”. By default the password length is 0 to 128 characters however the minimum password length must be configured to enforce a minimum of 15 characters. The upper limit can range from 15 characters or greater.

3.3.2.2 Guidance Documentation

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

a) identifies the characters that may be used in passwords and provides guidance to Security Administrators on the composition of strong passwords, and

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

Findings
PASS
[AGD] Section 4.3 describes the password complexity rules. This section includes the allowed character set which is consistent with the [ST]. This section also states that passwords may be 0 to 128 characters however in the evaluated configuration the minimum password length must be set to 15 characters.

3.3.2.3 Tests

139 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.

Findings
PASS
The evaluator confirmed that the TOE supports all claimed characters for passwords and that it is required to have a minimum length listed in the requirement. The minimum length was tested to show that when the minimum length was individually set to 15 and 8 the TOE did not accept passwords of length 14 and 7 and the TOE did accept passwords of length 15 and a password greater than 8, respectively.

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.

Findings
PASS
This test has been performed in conjunction with FIA_PMG_EXT.1 Test 1 by testing that the minimum length is enforced. Test 1 also confirmed that the TOE enforces the allowed characters.

3.3.3 FIA_UIA_EXT.1 User Identification and Authentication

3.3.3.1 TSS

140 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
PASS
Section 6.1 of the [ST] under heading FIA_UIA_EXT.1,FIA_UAU_EXT.2 provides the necessary information about how the TOE processes logging into the TOE. The TOE offers a local console CLI secured by a username and password, a remote CLI secured using either username/password or username/public key over SSH, and a remote web-based GUI operating over HTTPS secured using a username and password.

141 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
PASS
Section 6.1 of the [ST] under heading FIA_UIA_EXT.1,FIA_UAU_EXT.2 indicates "The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed, except for the login banner that is displayed prior to user authentication." This is consistent with the claims made in section 5.2 of the [ST].

142 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
Not Applicable: The TOE is not a distributed TOE.

143 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

Not Applicable: The TOE is not a distributed TOE.

3.3.3.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 4.1 describes how to configure administrator accounts to access the TOE. Section 3.3.2 and 3.3.3 describe the SSH and TLS configuration respectively to set the evaluated configuration. Section 3.3.2 includes the public key configuration for SSH connections. Additionally, section 4.4 describes how to configure the login banner.

3.3.3.3 Tests

145 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.

Findings
PASS
The evaluator confirmed that a Security Administrator is able to configure the appropriate credential supported for the login method. For that credential/login method, the user needs to provide correct I&A information to access the system; incorrect information results in denial of access.

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

Findings
PASS
The evaluator confirmed that no additional management interfaces were found, and only the login interface and banner are available to an external remote entity.

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.

Findings
PASS
The evaluator confirmed that viewing the banner is the only service available at the local console prior to authentication.

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.

Findings
Not Applicable: The TOE is not a distributed TOE.

3.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

146 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

147 None.

3.3.5.2 Guidance Documentation

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

Findings
PASS
[AGD] Does not specify any configuration needed to ensure authentication data is revealed. The evaluator confirm via the test below that this is in fact the case.

3.3.5.3 Tests

149 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.

Findings
PASS
The evaluator confirmed that the authentication information is obscured during the login process.

3.4 Security management (FMT)

3.4.1 FMT_MOF.1/ManualUpdate

3.4.1.1 TSS

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

Findings
Not applicable: The TOE is not a distributed TOE.

3.4.1.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 2.1 describes how an update to the TOE is performed. The TOE reboots when the upgrade is applied. This section warns the administrator that the TOE will reboot and cease to function when upgrading. Once the TOE reboots, it will be running the upgrade and resume operation.

152 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
Not applicable: The TOE is not a distributed TOE.

3.4.1.3 Tests

153 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.

Findings
PASS
The evaluator confirmed the TOE does not allow a non Security Administrator user to perform upgrades.

154 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.

Findings
PASS
This is covered by FPT_TUD_EXT.1 Test 1.

3.4.2 FMT_MTD.1/CoreData Management of TSF Data

3.4.2.1 TSS

155 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
PASS
The [ST] in section 6.1 (repeated throughout, such as in FIA_UIA_EXT.1, FMT_MTD.1/CoreData, etc) consistently indicates that no in-scope administrative functionality is available prior to login by an administrator. Note that the TOE banner can be displayed to end-users before establishing a login session as per FIA_UIA_EXT.1.

156 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
PASS
The [ST] in section 6.1 under heading FMT_MTD.1/CoreData states that only Authorized Administrators are capable of "uploading and enabling X509 certificates". The term "Authorized Administrator" is defined in the same section to denote any administrator who has been granted privileges to perform the relevant function. The TOE offers the ability to construct administrative users with a variety of different privilege level roles to aid in functional segregation. This functionality is claimed (for X.509 certificate construction) within FMT_MTD.1/CryptoKeys.

3.4.2.2 Guidance Documentation

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

Findings
PASS
All TSF-data-manipulating functions are covered throughout the remaining assurance activities. [AGD] Section 4.1 describes the administrative roles and the associated privileges. A user must have access granted and the appropriate role applied in order to access any of the functions.

158 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
PASS
[AGD] section 3.3.7 provides instructions for how to securely load a CA certificate into the trust store and designate it as a CA certificate.

3.4.2.3 Tests

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

3.4.3 FMT_SMF.1 Specification of Management Functions

160 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.

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

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

Findings
PASS

[ST] section 6.1 under FMT_SMF.1 lists all the management functions and indicates they are available via the CLI which is accessible via the console and SSH. A subset of the functions is available via the GUI.

[AGD] section 3 identifies the CLI (SSHv2 and local console) and GUI as the administrative interfaces.

While performing testing, the evaluator did not identify any additional administrative interfaces.

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

Findings

PASS

[ST] Section 1.2.2 and [AGD] 1.5 both describe the local administrative interface as a direct connection via the serial console port.

163 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

Not Applicable: The TOE is not a distributed TOE.

3.4.3.2 Guidance Documentation

164 See [ND-SD] section 2.4.4.1.

3.4.3.3 Tests

165 The evaluator tests management functions as part of testing the SFRs identified in [ND-SD] section 2.4.4. No separate testing for FMT_SMF.1 is required unless one of the management functions in FMT_SMF.1.1 has not already been exercised under any other SFR.

Findings

PASS

The following list maps the management function to the SFRs in which they were tested:

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

- Ability to administer the TOE locally and remotely;
  Covered by Test 1 in FIA_UIA_EXT.1
- Ability to configure the access banner;
  Covered by Test 1 in FTA_TAB.1
- Ability to configure the session inactivity time before session termination or locking;
  Covered by Test 1 in FTA_SSL.3
  Covered by Test 1 in FTA_SSL_EXT.1

- Ability to update the TOE, and to verify the updates using [hash comparison] capability prior to installing those updates;
  Covered by Test 1 in FPT_TUD_EXT.1 Test 1 and Test 3.
- Ability to configure the authentication failure parameters for FIA_AFL.1;
  Covered by Test 1 in FIA_AFL.1

[

- Ability to configure audit behaviour (e.g., changes to storage locations for audit; changes to behaviour when local audit storage space is full)
  Covered by Test 2 in FAU_STG_EXT.1
- Ability to manage the cryptographic keys;
  Covered by FAU_GEN.1; “Generating/import of cryptographic keys” and “Deleting of cryptographic keys”
- Ability to configure the cryptographic functionality;
  Covered by FAU_GEN.1; “Generating/import of cryptographic keys” and “Deleting of cryptographic keys” since only the cryptographic keys are configurable.
- Ability to re-enable an Administrator account;
  Covered by Test 1 in FIA_AFL.1
- Ability to set the time which is used for time-stamps;
  Covered by Test 1 in FPT_STM_EXT.1
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
  Covered by FAU_GEN.1; FIA_X509_EXT.1/Rev: Any addition of trust anchors in the TOE’s trust store and FIA_X509_EXT.1/Rev: Any removal of trust anchors in the TOE’s trust store
- Ability to import X.509v3 certificates to the TOE's trust store;
  Covered by FAU_GEN.1; FIA_X509_EXT.1/Rev: Any addition of trust anchors in the TOE’s trust store

]

3.4.4 FMT_SMR.2 Restrictions on security roles

3.4.4.1 TSS

166 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

PASS

[ST] section 6.1 under FMT_SMR.2 states that the “Authorized Administrator” includes privileged and semi-privileged administrator roles that can administer the TOE locally and remotely. An Authorized Administrator can create an unlimited number of user accounts and assign roles that are predefined by the system. This section defines each account role and lists the privileges granted for each role. Additionally, an Authorized Administrator cannot create user accounts with reserved names such as “operator” or “root”.

3.4.4.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 4.1 describes how to configure administrator accounts to access the TOE. Section 3.3.2 and 3.3.3 describe the SSH and TLS configuration respectively to set the evaluated configuration. Section 3.3.2 includes the public key configuration for SSH connections.

3.4.4.3 Tests

168 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
PASS
There are no explicit test activities and therefore none are recorded here. All interfaces are tested throughout the [DTR].

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

169 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
PASS
This information is found in the [ST] section 6.1 under heading FPT_APW_EXT.1 and FPT_SKP_EXT.1. When used in the evaluated configuration, there are no facilities to view plaintext passwords or keys. Use of specific CLI commands are necessary at setup time to ensure that such keys and passwords are stored using cryptographic means in the various configuration files.

3.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

3.5.2.1 TSS

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

Findings
PASS
This information is found in the [ST] section 6.1 under heading FPT_APW_EXT.1 and FPT_SKP_EXT.1. When used in the evaluated configuration, there are no facilities to view plaintext passwords or keys. Use of specific CLI commands are necessary at setup time to ensure that such keys and passwords are stored using cryptographic means in the various configuration files.

3.5.3 FPT_TST_EXT.1 TSF testing

3.5.3.1 TSS

171 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
PASS
The [ST] section 6.1 under heading FPT_TST_EXT.1 lists the self-tests which are run by the TSF. For each test, a description is provided in regard to what the test does and what constitutes the TSF successfully passing the test. Additionally, the final paragraph under this heading makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

172 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
Not Applicable: The TOE is not a distributed TOE.

3.5.3.2 Guidance Documentation

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

Findings

PASS
[AGD] Section 7 describes the modes of operation for the TOE. If an operational error occurs, the TOE reboots and enters booting mode. If a failure occurs during POST the system crashes and reports the appropriate information on the screen and saves it to the crashinfo file. The specific tests are described in section 3.3.9. The evaluator confirmed that these are consistent with the described tests in the TSS.

174 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
Not Applicable: The TOE is not a distributed TOE.

3.5.3.3 Tests

175 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.

176 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.

177 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.

178 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.

Findings
PASS
Note the production TOE does not log self-test status messages. The evaluator observed on an engineered build that the POST tests (self-tests) successfully ran on boot via the console output. The POST tests include the cryptographic tests and software integrity test defined in FPT_TST_EXT.1. The evaluator reviewed the vendor’s justification for the POST message satisfying all self-tests and determined it to be acceptable.

3.5.4 FPT_TUD_EXT.1 Trusted Update

3.5.4.1 TSS

179 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
PASS
Section 6.1 of the [ST] under FPT_TUD_EXT.1 describes that both the CLI and GUI can be used to query the currently active version of the TOE software/firmware. Updates to the TOE are not delayed.

180 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
PASS
Section 6.1 of the [ST] under FPT_TUD_EXT.1 describes that the TOE uses a published SHA-384 hash to validate the TOE firmware. The TOE verifies the hash before confirming that the new image can be installed.

181 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
Not applicable: Section 5.2 of the [ST] has not selected this option for FPT_TUD_EXT.1.2.

182 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
Not Applicable: The TOE is not a distributed TOE.

183 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
PASS
Section 6.1 of the [ST] in section FPT_TUD_EXT.1 indicates that automated updates are not allowed in the evaluated configuration. The Authorized Administrator must manually download the update.

3.5.4.2 Guidance Documentation

184 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
PASS
[AGD] Section 2.1 describes how the current TOE version can be queried. Updates to the TOE are not delayed.

185 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
PASS
[AGD] section 2.1 states that the TOE automatically compares the hash received against the computed hash. If there is a mismatch, the update is not installed.

186 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
PASS
[AGD] section 2.1 states that the hash is received via the configuration file.

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

Findings
Not Applicable: The TOE is not a distributed TOE.

188 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
Not Applicable: The TOE is not a distributed TOE.

189 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
Not Applicable: The TOE uses published hash for trusted updates.

3.5.4.3 Tests

190 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.

Findings
PASS
The evaluator confirmed the TOE displayed its current version, successfully installed a valid update, and displayed the updated version.

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.

Findings
Not Applicable: The ST does not claim digital signatures.

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 Security Administrator 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.

191 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.

Findings
PASS
The evaluator confirmed that the TOE correctly rejects an update when the hash verification fails.

Note: This was tested for an incorrect hash only. The TOE retrieves the update from the vendor's update server. The vendor is unable to host an update image without a hash value. Thus, updates provided to the TOE will always contain a hash and the first part of this AA is N/A.

192 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).

Note	The ST only claims manual updates which are covered in the tests above.

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

Findings
Not Applicable: The TOE is not a distributed TOE.

3.5.5 FPT_STM_EXT.1 Reliable Time Stamps

3.5.5.1 TSS

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

Findings
PASS
Section 6.1 in the [ST] under FPT_STM_EXT.1 provides the necessary information. The TOE time can be configured using the CLI and GUI. The TOE relies on the system clock provided by the underlying hardware on physical TOE devices. The system clock on virtual TOE devices is independent of ESXi.

NIAP TD0632

195 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.

Findings
Not Applicable: This selection is not made in [ST].

3.5.5.2 Guidance Documentation

196 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
PASS
[AGD] Section 4.6 describes how an authorized administrator can set the time on the TOE. The TOE does not use an NTP server.

NIAP TD0632

197 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.

Findings
PASS
[ST] States the time is synchronized with the underlying VS. [AGD] Does not specify any additional configuration needed by the TOE.

3.5.5.3 Tests

198 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.

Findings
PASS
The evaluator confirmed that the TOE supports direct setting of the time by the Security Administrator, and the Security Administrator was able to set the time on the TOE.

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.

Findings
Not Applicable: The TOE does not support NTP.

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.

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

199 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.

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

3.6 TOE Access (FTA)

3.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

3.6.1.1 TSS

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

Findings
PASS
Section 6.1 in the [ST] under FTA_SSL_EXT.1 and FTA_SSL.3 states that local termination is supported to the TOE. The setting can be set by the Authorized Administrator with the default value being 30 minutes.

3.6.1.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 4.5.2 specifies how to set the inactivity timeout period for the CLI.

3.6.1.3 Tests

202 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.

Findings
PASS
The evaluator confirmed that the TOE allows Security Administrator to configure several different values for the inactivity time period and observed that the session is terminated after the configured time period.

3.6.2 FTA_SSL.3 TSF-initiated Termination

3.6.2.1 TSS

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

Findings
PASS
Section 6.1 in the [ST] under FTA_SSL_EXT.1 and FTA_SSL.3 states that remote termination is supported to the TOE. CLI and GUI settings can be set individually by the Authorized Administrator with the default value being 30 minutes.

3.6.2.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 4.5.2 specifies how to set the inactivity timeout period for the CLI and GUI.

3.6.2.3 Tests

205 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.

Findings
PASS
The evaluator confirmed that the Security Administrator is able to configure several different values for the inactivity time period in each TOE component and to establish a remote interactive session. Additionally, they observed that the session is terminated after the configured time.

3.6.3 FTA_SSL.4 User-initiated Termination

3.6.3.1 TSS

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

Findings
PASS
Section 6.1 in the [ST] under FTA_SSL.4 states that the administrator is able to exit CLI and GUI sessions. To exit the CLI the ‘exit’ command is used and to exit the GUI the logout option in the drop-down menu is used.

3.6.3.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 4.5.3 states that sessions are terminated by clicking ‘exit’ in the GUI or typing ‘logout’ in the CLI.

3.6.3.3 Tests

208 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.

Findings
PASS
The evaluator confirmed that the interactive local session with the TOE is terminated when the 'Exit' CLI command is executed.

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.

Findings
PASS
The evaluator confirmed that a user is able to initiate an interactive remote session with the TOE and observed that the existing session is terminated when the user exits or logs off from the session.

3.6.4 FTA_TAB.1 Default TOE Access Banners

3.6.4.1 TSS

209 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
PASS
This information is provided in the [ST] in section 6.1 under FTA_TAB.1. The Authorized Administrator defines a custom login banner that will be displayed at the GUI and the CLI for both local and remote access configurations prior to allowing Authorized Administrator access through those interfaces.

3.6.4.2 Guidance Documentation

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

Findings
PASS
[AGD] Section 4.4 describes how the banner message is configured for all interfaces.

3.6.4.3 Tests

211 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.

Findings
PASS
The evaluator confirmed that the Security Administrator is capable of configuring a banner for the TOE and observed that the banner is displayed on each interface of the TOE.

3.7 Trusted path/channels (FTP)

3.7.1 FTP_ITC.1 Inter-TSF trusted channel

3.7.1.1 TSS

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

Findings
PASS
The TOE acts as a client for providing logging messages using SCP over SSHv2 as described in section 6.1 of the [ST] under FPT_ITC.1. In section 6.1 of the [ST] under FCS_SSHC_EXT.1, the TSS describes that the SCP connection to the remote entity is assured via the use of SSHv2 host keys (which employ public/private key cryptography). As there is only one trusted channel, it is trivial to map this function to the use of FCS_SSHC_EXT.1 SFR as described in section 6.1 of the [ST] under FCS_SSHC_EXT.1.

3.7.1.2 Guidance Documentation

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

Findings
PASS
[AGD] Sections 3.3.3 and 5 describe how to configure SSHv2 for remote logging. Section 3.3.3 also states that if the connection is unintentionally broken, the SSH client will need to re-authenticate to establish the connection again.

3.7.1.3 Tests

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

215 The evaluator shall perform the following tests:

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

Findings
PASS
The TOE maintains trusted channels to the remote audit log server which is set up as per the evaluated configuration. It is constantly tested throughout the evaluation.

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

Findings
PASS
The evaluator confirmed that the communication channel can be initiated from the TOE and the channel data is not sent in plaintext.

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

Findings
PASS
This is performed in conjunction with FTP_ITC.1 Test 2 which shows the channel data is not sent in plaintext.

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.

Findings
PASS
The evaluator confirmed that the TOE restored the connection from both short and long interruptions and that communications are appropriately protected, with no TSF data sent in plaintext.

216 Further assurance activities are associated with the specific protocols.

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

Findings
Not Applicable: The TOE is not a distributed TOE.

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

3.7.2 FTP_TRP.1/Admin Trusted Path

3.7.2.1 TSS

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

Findings
PASS
Section 6.1 of the [ST] under FTP_TRP.1/Admin reiterates that the TOE offers remote administrative capabilities over a remote CLI (via SSH) and over a remote Web UI (via TLS/HTTPS). The information in the TSS is consistent with the cryptographic channel claims made in section 5.2 of the [ST].

3.7.2.2 Guidance Documentation

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

Findings
PASS
[AGD] Sections 3.3.2 and 3.3.3 describe how to configure the TOE for remote SSH and TLS (HTTPS) connections respectively. Additionally, section 1.5 discusses the non-TOE components needed to establish a remote administrative session.

3.7.2.3 Tests

221 The evaluator shall perform the following tests:

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

Findings
PASS

The trusted paths are the TLS/HTTPS Web UI and SSH Remote CLI, which both are set up as per the evaluated configuration. They are constantly tested throughout the evaluation. TLS is tested in FCS_TLSS_EXT.1, and SSH is tested in FCS_SSHS_EXT.1.

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

Findings
PASS
The evaluator confirmed that the channel data is not sent in plaintext for each communication channel.

222 Further assurance activities are associated with the specific protocols.

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

Findings
Not Applicable: The TOE is not a distributed TOE.

4 Evaluation Activities for Optional Requirements

224 No optional requirements are claimed in the [ST].

5 Evaluation Activities for Selection-Based Requirements

5.1 Cryptographic Support (FCS)

5.1.1 FCS_HTTPS_EXT.1 HTTPS Protocol

5.1.1.1 TSS

225 The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818.

Findings
PASS
Section 6.1 of the [ST] under FCS_HTTPS_EXT.1 states that the TOE is conformant with RFC 2818. RFC2818 provides guidance that servers should implement an RFC-conformant version of TLS, that the server should be able to negotiate TLS (either using a distinct port or through a connection upgrade feature) and that the server should offer an appropriate certificate to ensure clients can confirm the identity. The TSS information in the [ST] section 6.1 clearly indicates that a TLS connection is used (which is reliant on claiming FCS_TLSS_EXT.1 and requires an RFC-conformant TLS implementation). The TSS in section 6.1 of the [ST] also provides that the server has an X.509 certificate to offer to remote clients.

5.1.1.2 Guidance Documentation

226 The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to configure TOE for use as an HTTPS client or HTTPS server.

Findings
PASS
The TOE only acts as an HTTPS server for remote administration. [AGD] Section 3.3.4 describes how to configure the TOE for remote administration via TLS/HTTPS.

5.1.1.3 Tests

227 This test is now performed as part of FIA_X509_EXT.1/Rev testing.

228 Tests are performed in conjunction with the TLS evaluation activities.

229 If the TOE is an HTTPS client or an HTTPS server utilizing X.509 client authentication, then the certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.

230

Note	FCS_HTTPS_EXT.1 is tested in conjunction with FIA_X509_EXT.1/Rev and FCS_TLSS_EXT.1. The TOE does not utilize X.509 client authentication.

5.1.2 FCS_SSHC_EXT.1 SSH Client

5.1.2.1 TSS

FCS_SSHC_EXT.1.2

NIAP TD0636

231 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
PASS
Section 6.1 of the [ST] under FCS_SSHC_EXT.1 identifies that the TOE uses rsa-sha2-256 for public key authentication and HMAC-SHA1 and HMAC-SHA256 for hashing. This is consistent with the FCS_CKM.1, FCS_COP.1/Hash and FCS_COP.1/SigGen claims.

NIAP TD0636

232 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
Not Applicable: FCS_SSHC_EXT.1.2 does not select password-based authentication.

FCS_SSHC_EXT.1.3

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

Findings
PASS
This information is provided in section 6.1 of the [ST] under FCS_SSHC_EXT.1. The TOE drops packets larger than 256KB.

FCS_SSHC_EXT.1.4

234 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
PASS

No optional characteristics are defined. Section 6.1 of the [ST] under FCS_SSHC_EXT.1 states that the TOE utilises aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr, and aes128-gcm@openssh.com for SSH encryption. These are identical to the claims made in the SFR in section 5.2 of the [ST].

FCS_SSHC_EXT.1.5

NIAP TD0636

235 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

PASS

Section 6.1 of the [ST] under FCS_SSHC_EXT.1 states that the TOE authenticates the SSH server key using a local database by associating the host name with its corresponding public key as described in RFC 4251 section 4.1.

NIAP TD0636

236 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

PASS

No optional characteristics are defined. Section 6.1 of the [ST] under FCS_SSHC_EXT.1 states that the TOE SSH client only negotiates rsa-sha2-256 during hostkey negotiation. This is identical to the claims made in the SFR in section 5.2 of the [ST].

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

Not applicable: No x509v3-based public key authentication algorithms are claimed.

FCS_SSHC_EXT.1.6

237 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

PASS

The integrity algorithms are described in section 6.1 of the [ST] under FCS_SSHC_EXT.1 as HMAC-SHA1 and HMAC-SHA256. This is consistent with the SFR in section 5.2.

FCS_SSHC_EXT.1.7

238 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

PASS

The key exchange algorithms are described in section 6.1 of the [ST] under FCS_SSHC_EXT.1 as diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521. This is consistent with the SFR in section 5.2.

FCS_SSHC_EXT.1.8

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

a) Both thresholds are checked by the TOE.

b) Rekeying is performed upon reaching the threshold that is hit first.

Findings

PASS

The [ST] in section 6.1 under FCS_SSHC_EXT.1 claims “[a] rekey occurs after a threshold of no longer than one hour and no more than one gigabyte of transmitted data.”

5.1.2.2 Guidance Documentation

NIAP TD0636

FCS_SSHC_EXT.1.2

NIAP TD0636

240 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

PASS

[AGD] Section 3.3.3 describes how to configure SSH. This section only describes how to configure the TOE for public key authentication. This is consistent with the [ST] claims.

FCS_SSHC_EXT.1.4

241 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

PASS

[AGD] Section 3.3.3 describes how to configure SSH. This section only instructs the user to configure the claimed algorithms.

FCS_SSHC_EXT.1.5

242 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

PASS

[AGD] Section 3.3.3 describes how to configure SSH. This section notes that the SSH client only negotiates rsa-sha2-256 which is consistent with the [ST] claim.

FCS_SSHC_EXT.1.6

243 The evaluator shall also check the guidance documentation to ensure that it contains instructions to the Security 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

PASS

[AGD] Section 3.3.3 describes how to configure SSH. This section notes that hmac-sha1 and hmac-sah256 are supported and the "None" MAC algorithm is not allowed. This is consistent with the [ST] claims.

FCS_SSHC_EXT.1.7

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

Findings

PASS

[AGD] Section 3.3.3 describes how to configure SSH. This section notes that the SSH client only negotiates diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 which is consistent with the [ST] claim.

FCS_SSHC_EXT.1.8

245 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
PASS
[AGD] Section 3.3.3 states the TOE enforces SSH rekey after one hour and/or after 1GB of data. This is not configurable.

5.1.2.3 Tests

FCS_SSHC_EXT.1.2

NIAP TD0636

246 Test objective: The purpose of these tests is to check the authentication of the client to the server using each claimed authentication method.

NIAP TD0636

247 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.

Findings
PASS
The evaluator confirmed that the TOE supports SSH-RSA public key and establish a successful SSH connection between the TOE and a custom server tool using public key authentication.

NIAP TD0636

248 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.

Findings
Not Applicable: Passwords are not claimed for FCS_SSHC_EXT.1.

FCS_SSHC_EXT.1.3

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

Findings
PASS

The evaluator confirmed the TOE rejects SSH packets larger than 256KB.

FCS_SSHC_EXT.1.4

250 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.

Findings
PASS
The evaluator confirmed that only claimed ciphers and cryptographic primitives are accepted to establish a SSH connection.

FCS_SSHC_EXT.1.5

251 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.

Findings
PASS
The evaluator confirmed that a user is able to establish an SSH connection using each of the public key algorithms specified by the requirement to authenticate an SSH server to the TOE.

252 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.

Findings
PASS
The evaluator confirmed that an SSH connection from the TOE to the SSH server with an unsupported public key algorithm is rejected.

FCS_SSHC_EXT.1.6

253 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.

254 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.

Findings
PASS
The evaluator confirmed that the TOE successfully establishes an SSH connection using the claimed algorithms.

255 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.

256 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 .

Findings
PASS
The evaluator confirmed that the TOE rejected an SSH connection using the hmac-sha1-96 algorithm, which is not claimed in the ST.

FCS_SSHC_EXT.1.7

257 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.

Findings
PASS
The evaluator confirmed that a connection from the TOE to the SSH server using each allowed key exchange method was successfully established.

FCS_SSHC_EXT.1.8

258 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.

259 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).

260 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.

Findings
PASS
Note the TOE does not maintain a persistent connection to the SSH syslog server to securely transfer logs. Instead, the TOE transfers the log files as scheduled or on-demand. Additionally, the log file sizes are configurable up to 104MB (maximum) and the log transfers occur much quicker than 1 hour over a single SSH connection. When successive log files are sent, they are sent over a new SSH connection. Using a custom SSH server, the evaluator performed a connection from the TOE client to connect to the server and confirmed a key operation and ensured that the TOE transfer succeeded in less than 1 hour. The evaluator then triggered another connection and ensured a new connection was established with a new keying operation. The evaluator transferred logs to the external syslog server and confirmed the TOE initiates a rekey for each log file sent. The evaluator also confirmed that the log files are transferred in under 1 hour which satisfies the required threshold.

261 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).

262 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).

263 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.

Findings
PASS
Note the TOE does not maintain a persistent connection to the SSH syslog server to securely transfer logs. Instead, the TOE transfers the log files as scheduled or on-demand. Additionally, the log file sizes are configurable up to 104MB (maximum) so the log transfers never reach the 1GB threshold over a single SSH connection. When successive log files are sent, they are sent over a new SSH connection. Using a custom SSH server, the evaluator performed a connection from the TOE client to connect to the server and confirmed a key operation and ensured that the TOE transfer succeeded and less than 1GB of data was exchanged. The evaluator then triggered another connection and ensured a new connection was established with a new keying operation. The evaluator confirmed the TOE initiates a rekey before 1 GB of data has been encrypted or decrypted using a key.

264 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).

Note	Neither threshold is configurable.

265 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

Note	There are no hardware limitations that affect the Traffic-Based Threshold rekey test.

FCS_SSHC_EXT.1.9

266 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 Security Administrator to accept or deny the key before continuing the connection.

Findings
PASS
The evaluator confirmed that the TOE rejected the connection due to an unknown host key.

267 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.

Findings
PASS
The evaluator confirmed that the TOE rejected the connection with the wrong host key.

5.1.3 FCS_SSHS_EXT.1 SSH Server

5.1.3.1 TSS

FCS_SSHS_EXT.1.2

NIAP TD0631

268 The evaluator shall check to ensure that the TSS contains a list of supported public key algorithms that are accepted for client 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
PASS
Section 6.1 of the [ST] under FCS_SSHS_EXT.1 states that the TOE uses rsa-sha2-256 and ecdsa-sha2-nistp256 for public key authentication. This is consistent with FCS_COP.1/SigGen.

NIAP TD0631

269 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
PASS
Section 6.1 of the [ST] under FCS_SSHS_EXT.1 states that the TOE compares the public key to the public key stored in the authorized_keys file. If the keys match, the connection is established.

NIAP TD0631

270 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
PASS
Section 6.1 of the [ST] under FCS_SSHS_EXT.1 states that the TOE itself supports password-based authentication. No other entities play a role in the authentication process.

FCS_SSHS_EXT.1.3

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

Findings
PASS
This information is provided in section 6.1 of the [ST] under FCS_SSHS_EXT.1. The TOE drops packets larger than 256KB.

FCS_SSHS_EXT.1.4

272 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
PASS
No optional characteristics are defined. Section 6.1 of the [ST] under FCS_SSHS_EXT.1 states that the TOE utilises aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr, and aes128-gcm@openssh.com for SSH encryption. These are identical to the claims made in the SFR in section 5.2 of the [ST].

FCS_SSHS_EXT.1.5

NIAP TD0631

273 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
PASS
Section 6.1 of the [ST] under FCS_SSHS_EXT.1 states that the TOE uses rsa-sha2-256 and ecdsa-sha2-nistp256 for public key authentication. This is consistent with the FCS_SSHS_EXT.1.5 claims.

FCS_SSHS_EXT.1.6

274 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
PASS
The integrity algorithms are described in section 6.1 of the [ST] under FCS_SSHS_EXT.1 as HMAC-SHA1 and HMAC-SHA2-256. This is consistent with the SFR in section 5.2.

FCS_SSHS_EXT.1.7

275 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
PASS
The key exchange algorithms are described in section 6.1 of the [ST] under FCS_SSHS_EXT.1 as diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384 and ecdh-sha2-nistp521. This is consistent with the SFR in section 5.2.

FCS_SSHS_EXT.1.8

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

a) Both thresholds are checked by the TOE.

b) Rekeying is performed upon reaching the threshold that is hit first.

Findings
PASS
The [ST] in section 6.1 under FCS_SSHS_EXT.1 claims the TOE "…will be rekeyed after a threshold of no longer than one hour, and no more than one gigabyte of transmitted data".

5.1.3.2 Guidance Documentation

FCS_SSHS_EXT.1.4

277 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
PASS
[AGD] Section 3.3.3 describes how to configure SSH. This section only instructs the user to configure the claimed algorithms.

FCS_SSHS_EXT.1.5

278 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
PASS
[AGD] Section 3.3.3 describes how to configure SSH. This section only describes how to configure the TOE public keys. This is consistent with the [ST] claims.

FCS_SSHS_EXT.1.6

279 The evaluator shall also check the guidance documentation to ensure that it contains instructions to the Security 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
PASS

[AGD] Section 3.3.3 describes how to configure SSH. This section notes that hmac-sha1 and hmac-sha256 are supported and the “None” MAC algorithm is not allowed. This is consistent with the [ST] claims.

FCS_SSHS_EXT.1.7

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

Findings
PASS
[AGD] Section 3.3.3 describes how to configure SSH. This section provides instructions such that only negotiates diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 are used.

FCS_SSHS_EXT.1.8

281 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
PASS
[AGD] Section 3.3.3 states the TOE enforces SSH rekey after one hour and/or after 1GB of data. This is not configurable.

5.1.3.3 Tests

FCS_SSHS_EXT.1.2

NIAP TD0631

282 Test objective: The purpose of these tests is to verify server supports each claimed client authentication method.

NIAP TD0631

283 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.

Findings

PASS
The evaluator confirmed that a user successfully logged into the TOE using an SSH client using the private key half.

NIAP TD0631

284 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.

Findings
PASS
The evaluator confirmed that the TOE rejected an SSH connection due to an unknown RSA private key half.

NIAP TD0631

285 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.

Findings
PASS
This test was conducted as part of FIA_AFL.1 Test 1.

NIAP TD0631

286 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.

Findings
PASS
This test was conducted as part of FIA_UIA_EXT.1 Test 1.

FCS_SSHS_EXT.1.3

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

Findings
PASS
The evaluator confirmed that the packet is dropped if the TOE receives a packet larger than 256k, as specified in the ST.

FCS_SSHS_EXT.1.4

288 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.

Findings
PASS
The evaluator confirmed that the TOE successfully established connections using the aes128-ctr, aes256-ctr, aes128-cbc, aes256-cbc, and aes128-gcm@openssh.com encryption algorithms. The evaluator also confirmed that the TOE offered no other algorithms.

FCS_SSHS_EXT.1.5

NIAP TD0631

289 Test objective: This test case is meant to validate that the TOE server will support host public keys of the claimed algorithm types.

NIAP TD0631

290 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.

Findings
PASS
The evaluator confirmed the TOE successfully identifies itself with each claimed host key algorithm and the connection succeeds.

NIAP TD0631

291 Has effectively been moved to FCS_SSHS_EXT.1.2.

NIAP TD0631

292 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.

NIAP TD0631

293 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.

Findings
PASS
The evaluator confirmed that the TOE failed to connect when a user attempted to force the use of an unsupported host public key algorithm.

FCS_SSHS_EXT.1.6

294 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.

295 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.

Findings
PASS
The evaluator confirmed that the TOE successfully established a connection with each claimed integrity algorithm. Note “implicit” is not specified for FCS_SSHS_EXT.1.6 in [ST].

296 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.

297 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.

Findings
PASS
The evaluator confirmed that the TOE failed to connect when a user attempted to force the use of an unsupported HMAC algorithm.

FCS_SSHS_EXT.1.7

298 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.

Findings
PASS
The evaluator confirmed that the TOE failed to connect with Diffie-Hellman Group1 SHA-1.

299 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.

Findings
PASS
The evaluator confirmed that the TOE successfully established connections with each claimed key exchange method.

FCS_SSHS_EXT.1.8

300 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.

301 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).

302 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.

Findings
PASS
The evaluator confirmed that the TOE initiated a rekey after reaching a threshold of one hour.

303 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).

304 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).

305 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.

Findings
PASS
The evaluator confirmed that the TOE initiated a rekey after reaching a threshold of one gigabyte of data.

306 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).

Note Neither threshold is configurable.

307 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.

Note There are no hardware limitations that affect the Traffic-Based Threshold rekey test.

5.1.4 FCS_TLSS_EXT.1 Extended: TLS Server Protocol

5.1.4.1 TSS

FCS_TLSS_EXT.1.1

308 The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component.

Findings

PASS

In the [ST] in section 6.1 for FCS_TLSS_EXT.1, the ciphersuites listed are identical to those listed in the SFR.

FCS_TLSS_EXT.1.2

309 The evaluator shall verify that the TSS contains a description of how the TOE technically prevents the use of old SSL and TLS versions.

Findings

PASS

Section 6.1 of the [ST] under FCS_TLSS_EXT.1 indicates that the TOE will not establish connections when SSLv2, SSLv3, or TLSv1.0 are received from the client. Only supported/configured TLS ciphersuites will be used. This is consistent with the claims made in the SFR in section 5.2.

FCS_TLSS_EXT.1.3

NIAP TD0635

310 If using ECDHE and/or DHE ciphers, the evaluator shall verify that the TSS lists all EC Diffie-Hellman curves and/or Diffie-Hellman groups used in the key establishment by the TOE when acting as a TLS Server. For example, if the TOE supports TLS_DHE_RSA_WITH_AES_128_CBC_SHA cipher and Diffie-Hellman parameters with size 2048 bits, then list Diffie-Hellman Group 14.

Findings
PASS
Section 6.1 of the [ST] in FCS_TLSS_EXT.1 describes that the claimed DHE ciphersuites use the standard parameters P, Q, and G for key exchange. This section also states that the secp256r1, secp384r1, and secp521r1 curves are used with TLS_ECDSA ciphers.

FCS_TLSS_EXT.1.4

311 The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246) and/or if session resumption based on session tickets is supported (RFC 5077).

312 If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption. The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.

313 If session tickets are supported, the evaluator shall verify that the TSS describes that session tickets adhere to the structural format provided in section 4 of RFC 5077 and if not, a justification shall be given of the actual session ticket format.

Findings
PASS
Section 6.1 of the [ST] in FCS_TLSS_EXT.1 states that the TOE supports session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2) and is enabled by default.

NIAP TD0569

314 If the TOE claims a (D)TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator verifies that the TSS describes how session resumption operates (i.e. what would trigger a full handshake, e.g. checking session status, checking Session ID, etc.). If multiple contexts are used the TSS describes how session resumption is coordinated across those contexts. In case session establishment and session resumption are always using a separate context, the TSS shall describe how the contexts interact with respect to session resumption (in particular regarding the session ID). It is acceptable for sessions established in one context to be resumable in another context.

Findings
PASS

Section 6.1 of the [ST] in FCS_TLSS_EXT.1 states that the TOE supports session resumption as a single context. The session is resumed using the session ID. If the client hello has a valid session ID, then the session resumes using that session ID.

5.1.4.2 Guidance Documentation

FCS_TLSS_EXT.1.1

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

Findings
PASS
[AGD] section 3.3.4 lists the ciphersuites enforced by default when the TOE is in FIPS mode. This list is consistent with [ST].

FCS_TLSS_EXT.1.2

316 The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

Findings
PASS
[AGD] Section 3.3.4 describes how to only configure TLSv1.1 and TLSv1.2.

FCS_TLSS_EXT.1.3

317 The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

Findings
PASS
[AGD] section 3.3.4 states that there are no specific parameters associated with the server key exchange. Hence, no configuration is necessary.

FCS_TLSS_EXT.1.4

NIAP TD0569

318 The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

Findings
PASS
[AGD] section 3.3.4 instructs the Authorized Administrator to enable TLS renegotiation when prompted: '*Would you like to Enable/Disable TLS Renegotiation for GUI HTTPS?<Y>'.*

5.1.4.3 Tests

FCS_TLSS_EXT.1.1

319 Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

Findings
PASS
The evaluator confirmed that the TOE successfully established a connection with the claimed ciphersuites.

320 Test 2: The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the server denies the connection.

Findings
PASS
The evaluator confirmed that the server denies the connection with unsupported ciphersuites and the TLS_NULL_WITH_NULL_NULL ciphersuite.

321 Test 3: The evaluator shall perform the following modifications to the traffic:

a) Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data.

Findings
PASS
The evaluator confirmed that the TOE rejected a connection when the client's finished handshake message was manipulated.

b) (Test Intent: The intent of this test is to ensure that the server's TLS implementation immediately makes use of the key exchange and authentication algorithms to: a) Correctly encrypt (D)TLS Finished message and b) Encrypt every (D)TLS message after session keys are negotiated.)

The evaluator shall use one of the claimed ciphersuites to complete a successful handshake and observe transmission of properly encrypted application data. The evaluator shall verify that no Alert with alert level Fatal (2) messages were sent.

The evaluator shall verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator shall examine the Finished message (encrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 11 22 33 44 55...) and confirm that it does not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c...), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. There is a chance that an encrypted Finished message contains a hexadecimal value of '14' at the position where a plaintext Finished message would contain the message type code '14'. If the observed Finished message contains a hexadecimal value of '14' at the position where the plaintext Finished message would contain the message type code, the test shall be repeated three times in total. In case the value of '14' can be observed in all three tests it can be assumed that the Finished message has indeed been sent in plaintext and the test has to be regarded as 'failed'. Otherwise it has to be assumed that the observation of the value '14' has been due to chance and that the Finished message has indeed been sent encrypted. In that latter case the test shall be regarded as 'passed'.

Findings

PASS

The evaluator confirmed that the TOE completed a successful handshake with one of the claimed ciphersuites and observed that the server finished message is encrypted.

FCS_TLSS_EXT.1.2

322 The evaluator shall send a Client Hello requesting a connection for all mandatory and selected protocol versions in the SFR (e.g. by enumeration of protocol versions in a test client) and verify that the server denies the connection for each attempt.

Findings

PASS

The evaluator confirmed that the TOE rejected negotiation with SSL 2.0, SSL 3.0, and TLS 1.0.

FCS_TLSS_EXT.1.3

323 Test 1: [conditional] If ECDHE ciphersuites are supported:

a) The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single supported elliptic curve specified in the Elliptic Curves Extension. The Evaluator shall verify (though a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange message and successfully establishes the connection.

Findings

PASS

The evaluator confirmed the TOE supports each claimed elliptic curve.

b) The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g. secp192r1 (0x13)) specified in RFC4492, chap. 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

Findings
PASS
The evaluator confirmed that the TOE failed to send back a Server Hello message when using a valid ECDHE ciphersuite with an unsupported curve, resulting in the termination of the connection.

324 Test 2: [conditional] If DHE ciphersuites are supported, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use a supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the message are the ones configured Diffie-Hellman parameter size(s).

Findings
PASS
The evaluator confirmed that the TLS client successfully connected to the TOE using a valid DHE ciphersuite, and the public key size returned in the Server Key Exchange message matches the expected bit size for the chosen DH parameter.

325 Test 3: [conditional] If RSA key establishment ciphersuites are supported, the evaluator shall repeat this test for each RSA key establishment key size. If any configuration is necessary, the evaluator shall configure the TOE to perform RSA key establishment using a supported key size (e.g. by loading a certificate with the appropriate key size). The evaluator shall attempt a connection using a supported RSA key establishment ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a certificate whose modulus is consistent with the configured RSA key size.

Findings
PASS
The evaluator confirmed that the TLS client successfully connected to the TOE using a valid RSA ciphersuite, and the TOE sent a certificate whose modulus is consistent with the configured RSA key size.

FCS_TLSS_EXT.1.4

Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).

NIAP TD0569

326 Remark: If multiple contexts are supported for session resumption, the session ID or session ticket may be obtained in one context for resumption in another context. It is possible that one or more contexts may only permit the construction of sessions to be reused in other contexts but not actually permit resumption themselves. For contexts which do not permit resumption, the evaluator is required to verify this behaviour subject to the description provided in the TSS. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

327 Test 1 [conditional]: If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test:

a) The client sends a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket.

b) The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake).

c) The client verifies the Server Hello message contains a zero-length session identifier or passes the following steps:

Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID.

d) The client completes the TLS handshake and captures the SessionID from the ServerHello.

e) The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d).

f) The client verifies the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

Findings
Not Applicable: Session resumption based on session IDs is supported by the TOE.

NIAP TD0569

328 Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ID may be obtained in one context for resumption in another context. There is no requirement that the session ID be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ID constructed in another context must be tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

329 Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

a) The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246).

Findings
PASS
The evaluator confirmed that the TOE resumed the session using the previous session ID.

b) The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

Findings
PASS
The evaluator confirmed that the TOE did not resume the session when modified session ID is used.

NIAP TD0569

330 Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ticket may be obtained in one context for resumption in another context. There is no requirement that the session ticket be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ticket constructed in another context must be tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

331 Test 3 [conditional]: If the TOE supports session tickets according to RFC5077, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

NIAP TD0556

a) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with an abbreviated handshake described in section 3.1 of RFC 5077 and illustrated with an example in figure 2. Of particular note: if the server successfully verifies the client's ticket, then it may renew the ticket by including a NewSessionTicket handshake message after the ServerHello in the abbreviated handshake (which is shown in figure 2). This is not required, however as further clarified in section 3.3 of RFC 5077.

Findings
Not Applicable: The TOE does not support session resumption using session tickets.

b) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.

Findings
Not Applicable: The TOE does not support session resumption using session tickets.

5.2 Identification and Authentication (FIA)

5.2.1 FIA_X509_EXT.1/Rev X.509 Certificate Validation

5.2.1.1 TSS

332 The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). It is expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected). It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).

Findings
PASS
Section 6.1 of the [ST] for FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3 indicates that the check of validity occurs at the time of import of the certificate. This is consistent with the fact that this TOE only offers a TLS server (which does not validate its own certificate except at import time).

333 The TSS shall describe when revocation checking is performed and on what certificates. If the revocation checking during authentication is handled differently depending on whether a full certificate chain or only a leaf certificate is being presented, any differences must be summarized in the TSS section and explained in the Guidance.

Findings
PASS
Section 6.1 of the [ST] for FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3 indicates that revocation checking occurs at the time of import of the certificate. The check is performed on all certificates in the chain where a CRL has been configured. CA certificates are checked whenever a new leaf certificate is loaded.

5.2.1.2 Guidance Documentation

334 The evaluator shall also ensure that the guidance documentation describes where the check of validity of the certificates takes place, describes any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) and describes how certificate revocation checking is performed and on which certificate.

Findings

PASS

[AGD] section 3.3.5 states that revocation is check via CRL as configured. Revocation checks are performed periodically (as configured) and at import for all certificates in the chain.

5.2.1.3 Tests

335 The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols:

a) Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store)

Findings

PASS

The evaluator confirmed that a certificate successfully uploads to the TOE with a valid chain. Then, after removing the intermediate CA from the trust store, the evaluator confirmed that the TOE rejected the same certificate due to a certificate signature verification failure.

Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.

Findings

PASS

This test was performed in conjunction with FIA_X509_EXT.1.1/Rev Test 1a above.

b) Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.

Findings

PASS

The evaluator confirmed that the TOE rejected the expired certificate.

c) Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.

Findings
PASS
The evaluator confirmed the TOE does not establish a connection when a certificate is revoked.

d) Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

Findings
PASS
The evaluator confirmed that the TOE does not accept a CRL status when the signing certificate does not have cRLsign key usage set.

e) Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)

Findings
PASS
The evaluator confirmed that the TOE rejects a certificate with a modified byte in the first eight bytes.

f) Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC5280 Sec. 4.1.1.3), which is normally the last field in the certificate, and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

Findings
PASS
The evaluator confirmed the TOE rejects a certificate with a modified signature.

g) Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.)

Findings
PASS
The evaluator confirmed the TOE rejects a certificate with a modified public key.

NIAP TD0527 (REVISED 1 December 2020)

336 The following tests are run when a minimum certificate path length of three certificates is implemented.

NIAP TD0527 (REVISED 1 December 2020)

h) Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall conduct the following tests:

Findings
Not Applicable: Tests 8a, 8b, and 8c are not applicable. FCS_COP.1/SigGen claims EC curves however, this in reference to the FCS_SSH*_EXT.* SFRs which do not use certificates. The TOE does not use EC certificates.

NIAP TD0527 (REVISED 1 December 2020)

Test 8a: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

Findings
Not Applicable: This test is not applicable. The TOE does use EC certificates.

NIAP TD0527 (REVISED 1 December 2020)

Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

Findings

Not Applicable: This test is not applicable. The TOE does use EC certificates.

NIAP TD0527 (REVISED 1 December 2020)

Test 8c: The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.

Findings
Not Applicable: This test is not applicable. The TOE does use EC certificates.

337 The evaluator shall perform the following tests for FIA_X509_EXT.1.2/Rev. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.

338 The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation).

339 For each of the following tests the evaluator shall create a chain of at least three certificates: a self-signed root CA certificate, an intermediate CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

a) Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE’s trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

Findings
PASS
The evaluator confirmed that the TOE failed to load a CA certificate into the trust store due to the absence of the basicConstraints extension.

b) Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

Findings
PASS
The evaluator confirmed that the TOE failed to load a CA certificate into the trust store which has a basicConstraints of False extension.

340 The evaluator shall repeat these tests for each distinct use of certificates. Thus, for example, use of certificates for TLS connection is distinct from use of certificates for trusted updates so both of these uses would be tested. But there is no need to repeat the tests for each separate TLS channel in FTP_ITC.1 and FTP_TRP.1/Admin (unless the channels use separate implementations of TLS).

Note The distinct uses of certificates are covered in the tests above.

5.2.2 FIA_X509_EXT.2 X.509 Certificate Authentication

5.2.2.1 TSS

341 The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.

Findings
PASS
The TSS claims in section 6.1 of the [ST] for FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3 that the administrative user manually installs and selects the certificate used by the TOE for each certificate. The [AGD] sections 3.3.4, 3.3.5, and 3.3.6 provide the necessary guidance to administrators to configure certificates.

342 The evaluator shall examine the TSS to confirm that it describes the behaviour of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the guidance documentation contains instructions on how this configuration action is performed.

Findings
PASS
Section 6.1 of the [ST] for FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3 states that if the connection to determine the certificate validity cannot be established, the TOE will accept the certificate based on the last known state.

5.2.2.2 Guidance Documentation

343 The evaluator shall also ensure that the guidance documentation describes the configuration required in the operating environment so the TOE can use the certificates. The guidance documentation shall also include any required configuration on the TOE to use the certificates. The guidance document shall also describe the steps for the Security Administrator to follow if the connection cannot be established during the validity check of a certificate used in establishing a trusted channel.

Findings
🟩 PASS
[AGD] Sections 3.3.5 through 3.3.8 describe how to configure the TOE to use certificates. Section 3.3.5 states that if a connection cannot be established during the validity check, the TOE will accept the certificate.

5.2.2.3 Tests

344 The evaluator shall perform the following test for each trusted channel:

345 The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the guidance documentation to determine that all supported administrator-configurable options behave in their documented manner.

Findings
🟩 PASS
The evaluator confirmed when the TOE is unable to fetch revocation status it accepts the certificate.

5.2.3 FIA_X509_EXT.3 Extended: X509 Certificate Requests

5.2.3.1 TSS

346 If the ST author selects "device-specific information", the evaluator shall verify that the TSS contains a description of the device-specific fields used in certificate requests.

Findings
Not Applicable: FIA_X509_EXT.3.1 does not select “device-specific information”.

5.2.3.2 Guidance Documentation

347 The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certificate Request. If the ST author selects "Common Name", "Organization", "Organizational Unit", or "Country", the evaluator shall ensure that this guidance includes instructions for establishing these fields before creating the Certification Request.

Findings
PASS
[AGD] section 3.3.5 states the when creating a CSR, the name, country, organization name, organizational unit, and email (optional) should be included. This is consistent with [ST].

5.2.3.3 Tests

348 The evaluator shall perform the following tests:

a) Test 1: The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.

Findings
PASS
The evaluator confirmed that the TOE requires the necessary information when constructing a new X.509 CSR.

b) Test 2: The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message and demonstrate that the function succeeds.

Findings
PASS
The evaluator confirmed the TOE fails to load a signed CSR (certificate) when it cannot complete the trust chain and successfully loads a signed CSR when it is able to complete the trust chain.

5.3 Security management (FMT)

5.3.1 FMT_MOF.1/Functions Management of security functions behaviour

5.3.1.1 TSS

349 For distributed TOEs see [ND-SD] chapter 2.4.1.1.

Findings
Not Applicable: The TOE is not a distributed TOE.

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

Findings
PASS
Section 6.1 of the [ST] for FMT_MOF.1/Functions states that the TOE provides administrative user with a CLI and web-based GUI to interact with and manage the security functions of the TOE. The evaluator determined that this includes the FMT_MOF.1/Functions which includes the ability to determine the behaviour of the transmission of audit data to an external IT entity.

5.3.1.2 Guidance Documentation

351 For distributed TOEs see [ND-SD] chapter 2.4.1.2.

Findings
Not Applicable: The TOE is not a distributed TOE.

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

Findings
PASS
[AGD] section 5 states that the audit handling configuration is performed once when the device is initially configured and provides instructions for the Authorized Administrator to view the configuration settings.

5.3.1.3 Tests

353 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.

Findings
Not Applicable: The ST does not claim this functionality and this test need not be conducted.

354 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.

355 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.

Findings
Not Applicable: The ST does not claim this functionality and this test need not be conducted.

356 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.

Findings
Not Applicable: The ST does not claim this functionality and this test need not be conducted.

357 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.

358 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.

Findings
Not Applicable: The ST does not claim this functionality and this test need not be conducted.

359 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.

Findings
Not Applicable: The ST does not claim this functionality and this test need not be conducted.

360 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.

361 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.

Findings
Not Applicable: The ST does not claim this functionality and this test need not be conducted.

362 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.

Findings
PASS
The evaluator confirmed that a user without administrator privileges is not allowed to transmit audit data to an external IT entity.

363 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 Security Administrator authentication shall be successful.

Findings
PASS

The previous test case combines the above test case with the test case for unprivileged users.

5.3.2 FMT_MTD.1/CryptoKeys Management of TSF Data

5.3.2.1 TSS

364 For distributed TOEs see [ND-SD] chapter 2.4.1.1.

Findings
Not Applicable: The TOE is not a distributed TOE.

365 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
PASS
Section 6.1 of the [ST] for FMT_MTD.1/CryptoKeys lists the keys that can be managed by Authorized Administrators of the TOE: • X.509v3 certificates: import, modify, delete • SSH host keys: generate, modify, delete • SSH user keys: generate, delete

5.3.2.2 Guidance Documentation

366 For distributed TOEs see [ND-SD] chapter 2.4.1.2.

Findings
Not Applicable: The TOE is not a distributed TOE.

367 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
PASS
[AGD] section 3.3.3 provides instructions for managing the SSH user and host keys and section 3.3.5 describes management of X.509v3 certificates. The functions covered are consistent with [ST].

5.3.2.3 Tests

368 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.

Findings
PASS The evaluator confirmed that a non-privileged user cannot generate a new public/private key pair.

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

Findings
PASS Certificate configuration is performed as part of FIA_X509_EXT.3.

6 Evaluation Activities for Security Assurance Requirements

6.1 ASE: Security Target

6.1.1 General ASE

370 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
PASS
The ASE CEM work units are documented in the proprietary ETR. The TSS Evaluation Activities are documented throughout this report.

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

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

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

6.2 ADV: Development

6.2.1 Basic Functional Specification (ADV_FSP.1)

372 The EAs for this assurance component focus on understanding the interfaces (e.g., application programming interfaces, command line interfaces, graphical user interfaces,

network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in Section 2, and in EAs for AGD, ATE and AVA SARs in other parts of Section 5.

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

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

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

6.2.1.1 Evaluation Activity

376 *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.*

377 In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g. audit review or performing updates). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. The intent is that these interfaces will be adequately tested and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied.

378 The set of TSFI that are provided as evaluation evidence are contained in the Administrative Guidance and User Guidance.

Findings
PASS
From section 7.2.1 of the [NDcPP]: "For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation." The [ST] and the guidance documentation comprise the functional specification. The evaluator was able to perform the Evaluation Activities specified in the [ND-SD], so the evaluator concluded that the functional specification sufficiently describes the parameters, purpose, and method of use for each TSFI that is identified as being security relevant.

6.2.1.2 Evaluation Activity

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

Findings
PASS
Please see the previous work unit.

6.2.1.3 Evaluation Activity

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

381 The evaluator uses the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2, including the EAs associated with testing of the interfaces.

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

383 However, if the evaluator is unable to perform some other required EA because there is insufficient design and interface information, then the evaluator is entitled to conclude that an adequate functional specification has not been provided, and hence that the verdict for the ADV_FSP.1 assurance component is a 'fail'.

Findings
PASS
From section 7.2.1 of the [NDcPP]: "For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation." The [ST] and the guidance documentation comprise the functional specification. The interfaces are implicitly mapped to SFRs if they are used to satisfy an Evaluation Activity for a specific SFR. The evaluator was able to perform the Evaluation Activities specified in the [ND-SD]; the Findings for SFR related Evaluation Activities are the mapping of interfaces to SFRs.

6.3 AGD: Guidance Documents

384 It is not necessary for a TOE to provide separate documentation to meet the individual requirements of AGD_OPE and AGD_PRE. Although the EAs in this section are described under the traditionally separate AGD families, the mapping between the documentation provided by the developer and AGD_OPE and AGD_PRE requirements may be many-to-many, as long as all requirements are met in documentation that is delivered to Security Administrators and users (as appropriate) as part of the TOE.

385 Note that additional Evaluation Activities for the guidance documentation in the case of a distributed TOE are defined in section A.9.1.1. (in the [ND-SD])

6.3.1 Operational User Guidance (AGD_OPE.1)

386 The evaluator performs the CEM work units associated with the AGD_OPE.1 SAR. Specific requirements and EAs on the guidance documentation are identified (where relevant) in the individual EAs for each SFR.

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

6.3.1.1 Evaluation Activity

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

Findings
PASS
The guidance documentation is posted to the NIAP website, ensuring administrators are aware of the documentation.

6.3.1.2 Evaluation Activity

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

Findings
PASS
There is only one operational environment claimed in [ST] section 1.4. All TOE platforms claimed in [ST] are covered by the operational guidance. This is evidenced by the platform equivalency.

6.3.1.3 Evaluation Activity

390 *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
PASS
[AGD] sections 3.2.1, 3.3.3, and 3.3.4 describe the configuration of the cryptographic operations (i.e., engines, protocols, algorithms, key sizes) to be consistent with the evaluated configuration. The [AGD] provides instructions for ensuring the evaluated functionality is used.

6.3.1.4 Evaluation Activity

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

Findings
PASS
[AGD] sections 1.5 and 1.6 clarify the evaluated functionality. The evaluator confirmed [AGD] covers configuration of the in-scope functionality where additional configuration might be required.

6.3.1.5 Evaluation Activity

392 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
PASS
See section 6.3.1.3 for configuration of the cryptographic engine. [AGD] section 2.1 provides a description of the update process. See section 6.3.1.4 for details as to what was covered by the EAs.

6.3.2 Preparative Procedures (AGD_PRE.1)

393 The evaluator performs the CEM work units associated with the AGD_PRE.1 SAR. Specific requirements and EAs on the preparative documentation are identified (and where relevant are captured in the Guidance Documentation portions of the EAs) in the individual EAs for each SFR.

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

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

6.3.2.1 Evaluation Activity

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

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

Findings
PASS
The operational user guidance describes the security measured to be followed to fulfill the security objectives for the operational environment.

6.3.2.2 Evaluation Activity

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

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

6.3.2.3 Evaluation Activity

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

Findings
PASS
See previous work unit.

6.3.2.4 Evaluation Activity

400 *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
PASS

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

6.3.2.5 Evaluation Activity

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

402 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

PASS

The entire [AGD] document is designed to ensure the administrator is aware of how to configure the TOE to provide a protected administrative capability.

The TOE has default TOE passwords. However, the [AGD] instructs the administrator to change the password to meet the minimum password requirements as stated in the [ST]. These complexity requirements are enforced by the TOE rather than by policy.

6.4 ALC: Life-cycle Support

6.4.1 Labelling of the TOE (ALC_CMC.1)

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

Findings

PASS

The evaluator verified that the ST, TOE and Guidance are all labelled with the same hardware versions and software. The information is specific enough to procure the TOE and it includes hardware models and software versions. The evaluator checked the TOE software version and hardware identifiers during testing by examining the actual machines used for testing.

6.4.2 TOE CM coverage (ALC_CMS.1)

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

Findings

PASS

The evaluator verified that the ST, TOE and Guidance are all labelled with the same hardware versions and software. The information is specific enough to procure the TOE and it includes hardware models and software versions. The evaluator checked the TOE software version and hardware identifiers during testing by examining the actual machines used for testing.

6.5 ATE: Tests

6.5.1 Independent Testing – Conformance (ATE_IND.1)

405 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.

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

407 The evaluator should consult Appendix 709 when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

408 Note that additional Evaluation Activities relating to evaluator testing in the case of a distributed TOE are defined in [ND-SD] section A.9.3.1.

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

6.6 Vulnerability Assessment

6.6.1 Vulnerability Survey (AVA_VAN.1)

409 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.

410 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.

411 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.

6.6.1.1 Evaluation Activity (Documentation)

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

413 *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

414 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
PASS
The evaluator collected this information from the developer which was used to feed into the Type 1 Flaw Hypotheses search (below).

415 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 [ND-SD] 3.4.1.2 and 3.5.1.2.

Findings
PASS
The TOE is not a distributed TOE.

6.6.1.2 Evaluation Activity

416 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

PASS

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

- CVEs
    - NIST National Vulnerabilities Database (can be used to access CVE and US-CERT databases identified below): https://web.nvd.nist.gov/view/vuln/search
    - Common Vulnerabilities and Exposures: http://cve.mitre.org/cve/
    - Common Vulnerabilities and Exposures: https://www.cvedetails.com/vulnerability-search.php
- US-CERT: http://www.kb.cert.org/vuls/html/search
- Tenable Network Security: https://www.tenable.com/cve
- Tipping Point Zero Day Initiative: http://www.zerodayinitiative.com/advisories
- Offensive Security Exploit Database: https://www.exploit-db.com/
- Rapid7 Vulnerability Database: https://www.rapid7.com/db/vulnerabilities
- Cisco Security Advisory: https://sec.cloudapps.cisco.com/security/center/publicationListing.x#~FilterByProduct

Type 1 Hypothesis searches were conducted on August 19, 2024 and included the following search terms:

- Cisco C195
- Cisco C395
- Cisco C695
- Cisco C695F
- Cisco C100v
- Cisco C300v
- Cisco C600v
- UCS-C220-M5
- UCS-C220-M6
- UCS-C240-M5
- UCS-C480-M5
- UCS-C240-M6
- Cisco ESA
- Cisco Email Security Appliance
- Cisco AsyncOS
- Intel Xeon Gold 6126
- Intel Xeon Gold 6342
- Intel Xeon Silver 4116
- Intel Xeon Silver 4110
- Intel Xeon Gold 6248r
- Intel Xeon Platinum 8360Y
- OpenSSH
- CiscoSSL FOM
- OpenSSL
- FreeBSD
- CiscoSSH

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

The evaluation team evaluated Type 2 flaw hypotheses in accordance with [ND-SD] sections A.1.2 and A.5, and no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.

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

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