# NETSCOUT

## nGenius 5000 & 7000 Series Packet Flow Switches with PFOS 6.0.6

# Assurance Activity Report

**Version** 0.14

June 2022

**Document prepared by**


www.lightshipsec.com

# Table of Contents

# 1 Introduction

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

## 1.1 Evaluation Identifiers

**Table 1: Evaluation Identifiers**

| | |
|---|---|
| **Scheme** | Canadian Common Criteria Scheme |
| **Evaluation Facility** | Lightship Security |
| **Developer/Sponsor** | NETSCOUT Systems, Inc. |
| **TOE** | nGenius 5000 & 7000 Series Packet Flow Switches with PFOS 6.0.6 Build: 6.0.6.4 |
| **Security Target** | nGenius 5000 & 7000 Series Packet Flow Switches with PFOS 6.0.6 Security Target, v1.7 |
| **Protection Profile** | collaborative Protection Profile for Network Devices, v2.2E (NDcPP), 23-March-2020 |

## 1.2 Evaluation Methods

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

**Table 2: Evaluation Methods**

| | |
|---|---|
| **Evaluation Criteria** | CC v3.1R5 |
| **Evaluation Methodology** | CEM v3.1R5 |
| **Supporting Documents** | Evaluation Activities for Network Device cPP, v2.2 (NDcPP-SD) |
| **Interpretations** | **NDcPP v2.2e**<br>TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1)<br>TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4<br>TD0536: NIT Technical Decision for Update Verification Inconsistency<br>TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3 |

| |
|---|
| TD0538: NIT Technical Decision for Outdated link to allowed-with list |
| TD0546: NIT Technical Decision for DTLS - clarification of Application Note 63 |
| TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN |
| TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test |
| TD0556: NIT Technical Decision for RFC 5077 question |
| TD0563: NiT Technical Decision for Clarification of audit date information |
| TD0564: NiT Technical Decision for Vulnerability Analysis Search Criteria |
| TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7 |
| TD0570: NiT Technical Decision for Clarification about FIA_AFL.1 |
| TD0571: NiT Technical Decision for Guidance on how to handle FIA_AFL.1 |
| TD0572: NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers |
| TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e |
| TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3 |
| TD0591: NIT Technical Decision for Virtual TOEs and hypervisors |
| TD0592: NIT Technical Decision for Local Storage of Audit Records |
| TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server |
| TD0632: NIT Technical Decision for Consistency with Time Data for vNDs |
| TD0633: NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance |
| TD0634: NIT Technical Decision for Clarification required for testing IPv6 |

| | |
|---|---|
| | TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters |
| | TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH |

## 1.3 Reference Documents

**Table 3: List of Reference Documents**

| Ref | Document |
|---|---|
| [ST] | nGenius 5000 & 7000 Series Packet Flow Switches with PFOS 6.0.6 Security Target, v1.7, June 2022 |
| [PP] | collaborative Protection Profile for Network Devices, Version 2.2e |
| [SD] | Supporting Document Mandatory Technical Document Evaluation Activities for Network Device cPP, v2.2 |
| [AGD] | NETSCOUT nGenius 5000 & 7000 Series Packet Flow Switches with PFOS 6.0.6 Common Criteria Guide, v1.1, June 2022 |
| [USER] | NETSCOUT Packet Flow Operating Software (PFOS) 6.x User Guide Software Version 6.0.6, 733-1485 / December 2021 |
| [INSTALL] | nGenius® PFS 5000 Series Packet Flow Switches Quick Connection Guide<br>PFOS Installation Guide for Qualified PFS Devices |
| [CLI] | NETSCOUT Packet Flow Operating Software (PFOS) 6.x CLI Reference Guide Software Version 6.0.6, 733-1486 / December 2021 |
| [REL] | NETSCOUT Packet Flow Operating Software (PFOS) 6.x Release Notes Software Version 6.0.6, 733-1488 / April 2022 / Revision C |

# 2 Evaluation Activities for SFRs

## 2.1 Security Audit (FAU)

### 2.1.1 FAU_GEN.1 Audit data generation

#### 2.1.1.1 TSS

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

| | |
|---|---|
| **Findings:** | [ST] 6.1.1 states: "The following information is logged as a result of the Security Administrator generating/importing or deleting cryptographic keys:<br><br>a) Generate SSH key-pair. Action and key reference.<br>b) Generate cryptographic keys. Action and key reference.<br>c) Import Certificate. Action and key reference.<br>d) Import CA Certificate. Action and key reference." |

4 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:** | This is not a distributed TOE. |

#### 2.1.1.2 Guidance Documentation

5 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:** | Section 3.12 'Events' of the [AGD] contains a table of example auditable events as required by FAU_GEN.1 and the SFRs claimed in the [ST]. |

6 The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

| | |
|---|---|
| **Findings:** | The evaluator performed this activity as part of those AAs associated with ensuring the corresponding guidance documentation satisfied their independent requirements. However, overall, the evaluator considered the administrator guides published by the vendor. The evaluator reviewed the contents of the documentation and looked specifically for functionality related to the scope of the evaluation. Where there was missing or incomplete descriptions for the functionality such that the user could not complete the testing AAs, the evaluator requested the vendor to supply augmented guidance information. In the end, the vendor provided a more comprehensive guidance document in the form of [USER]. However, AAs for guidance were performed combining all the Guidance documents listed in the **Table 3: List of Reference Documents** of this document. |

#### 2.1.1.3 Tests

7 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.

8 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.

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

| **High-Level Test Description** |
|---|
| The evaluator confirmed the TOE produces all of the required audit events in conjunction with the applicable test activity and verified that it included all of the required fields and matched the reference samples provided in [AGD].<br><br>The TOE is not a distributed TOE. |
| Findings: PASS |

### 2.1.2 FAU_GEN.2 User identity association

#### 2.1.2.1 Tests

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

11 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.

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

### 2.1.3 FAU_STG_EXT.1 Protected audit event storage

#### 2.1.3.1 TSS

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

| | |
|---|---|
| **Findings:** | [ST] 6.1.3 states: "The Security Administrator can configure the TOE to send logs to a Syslog server. Log events are sent in real-time. Logs are sent via SSH as described by FCS_SSHC_EXT.1." |

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

| | |
|---|---|
| **Findings:** | [ST] 6.1.3 states the following: "The amount of audit data that may be stored locally is dependent on the available disk space, which is 4GB.<br><br>When the local audit data store is full, the TOE will overwrite audit records starting with the oldest audit record.<br><br>Only authorized administrators may view audit records and no capability to modify the audit records is provided." |

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

| | |
|---|---|
| **Findings:** | The TOE is a standalone TOE. [ST] 6.1.3 indicates audit data may be stored locally or sent to an external Syslog server via SSH. |

15 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:** | Section 6.1.3 of the [ST] states: "When the local audit data store is full, the TOE will overwrite audit records starting with the oldest audit record." |

16 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:** | Section 6.1.3 of the [ST] states: "Log events are sent in real-time." |
|---|---|

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

| **Findings:** | The TOE is not a distributed TOE. |
|---|---|

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

| **Findings:** | The TOE is not a distributed TOE. |
|---|---|

### 2.1.3.2 Guidance Documentation

19 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:** | Section 3.7 'Audit Logging' of the [AGD] points to the Configuring the System and Ports->Configuring System Settings->System Settings->Syslog->Send System Logs to Remote Server over SSH Tunnel section of [USER] to configure a trusted channel to the audit server via SSH including the configuration selections that need to be made. |
|---|---|

20 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:** | Section 3.7 'Audit Logging' of the [AGD] states that the TOE stores and sends audit data in real-time. When the local log's space is full, the TOE will overwrite the oldest logs. |
|---|---|

21 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:** | In the evaluated configuration the TOE only handles the audit log being full by overwriting the oldest records. No other configuration is supported. Section 3.7 'Audit Logging' of [AGD] states how the TOE handles audit records when the local space is full. |
|---|---|

### 2.1.3.3 Tests

22 Testing of the trusted channel mechanism for audit will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall perform the following additional tests for this requirement:

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

| **High-Level Test Description** |
|---|
| This test is performed in conjunction with FTP_ITC.1. |
| Findings: PASS |

b) Test 2: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.3. Depending on the configuration this means that the evaluator has to check the content of the audit data when the audit data is just filled to the maximum and then verifies that

1) The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option 'drop new audit data' in FAU_STG_EXT.1.3).

2) The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option 'overwrite previous audit records' in FAU_STG_EXT.1.3)

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

| **High-Level Test Description** |
|---|
| The maximum audit records held by the TOE are 1000 events. The evaluator observed the oldest audit event then performed actions to generate additional audits. Once the threshold of 1000 events had been exceeded, the evaluator queried the audit log and confirmed that the oldest events had been overwritten. |
| Findings: PASS |

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

| **Findings:** | The ST does not claim this functionality for the TOE. |
|---|---|

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.

| **Findings:** | The TOE is not a distributed TOE. |
|---|---|

# 2.2 Cryptographic Support (FCS)

## 2.2.1 FCS_CKM.1 Cryptographic Key Generation

### 2.2.1.1 TSS

23 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:** | Section 6.2.1 of the [ST] states ECC P-256, P-384, P-521 are used in SSH and TLS. Key sizes for each scheme are 256, 384 and 512 respectively. |
|---|---|

### 2.2.1.2 Guidance Documentation

24 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:** | Section 3.4 of [AGD] specifies that the TOE should be configured into FIPS mode. After the TOE is configured into FIPS mode, no further configuration is needed to support the claimed key generation scheme(s) and key size(s). |
|---|---|

### 2.2.1.3 Tests

25 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**

26 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*.

27 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

28 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*

29 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*

30 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)***

31 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.

32 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

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

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

34 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.

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

36 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.

37 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

38 for each FFC parameter set and key pair.

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

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

| **Findings:** | The vendor uses CAVP certificates C1880, C1881, and A1882 for EC key generation. These are described in the Security Target in Table 4. |
|---|---|

## 2.2.2 FCS_CKM.2 Cryptographic Key Establishment

### 2.2.2.1 TSS

40 [Modified by TD0580] 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.

41 [Modified by TD0580] The intent of this activity is to be able to identify the scheme being used by each service. This would mean, for example, one way to document scheme usage could be:

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

42 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:** | Table 13 in [ST] section 6.2.2 illustrates the various key establishment schemes and associated usage/service. These are consistent with the key generation mechanisms described in the FCS_CKM.1.1 component in the [ST]. |

#### 2.2.2.2 Guidance Documentation

43 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:** | Section 3.4 of [AGD] specifies that the TOE should be configured into FIPS mode. After the TOE is configured into FIPS mode, no further configuration is needed to support the claimed key establishment scheme(s). |

#### 2.2.2.3 Tests

[Modified by TD0580]

**Key Establishment Schemes**

44 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***

45 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*

46 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.

47 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.

48 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.

49 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.

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

*Validity Test*

51 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.

52 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).

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

***RSA-based key establishment schemes***

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

***FFC Schemes using "safe-prime" groups***

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

| | |
|---|---|
| **Findings:** | The vendor uses CAVP certificates C1880, C1881, and A1882 for EC key establishment. These are described in the Security Target in Table 4. |

## 2.2.3 FCS_CKM.4 Cryptographic Key Destruction

### 2.2.3.1 TSS

56 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:** | Section 6.2.3 of the [ST] states: "Keys held in volatile memory are zeroized after use by overwriting the key storage area with zeroes. Keys held in flash memory may be destroyed using a Command Line Interface (CLI) command to overwrite the entire flash memory an administrator specified number of times (between 1 and 10) with zeroes."<br><br>Furthermore, section 6.2.3 of the [ST] states: "Table 15 shows the origin, storage location and destruction details for cryptographic keys and passwords. Unless otherwise stated, the keys are generated by the TOE."<br><br>Upon inspection of Table 15 found in section 6.5.1 of the [ST], the origin, storage location and destruction method for each key are clearly described. Keys are stored in plaintext in persistent Flash and volatile RAM on the device. Destruction methods indicated by Table 15 are consistent with the selections made in the FCS_CKM.4 SFR in section 5.3 of the [ST]. |

57 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:** | Section 6.2.3 of the [ST] states: "Keys held in flash memory may be destroyed using a Command Line Interface (CLI) command to overwrite the entire flash memory an administrator specified number of times (between 1 and 10) with zeroes." |

58 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:** | The [ST] does not make this selection. |

59 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:** | The TSS does not identify any keys stored in a non-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.

60 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:** | The TSS does not identify any configurations or circumstances where the TOE does not conform to the described key destruction method. |
|---|---|

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

| **Findings:** | The [ST] does not make this selection. |
|---|---|

#### 2.2.3.2 Guidance Documentation

62 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.

63 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:** | Section 6.2.3 and Table 15 of the [ST] discusses the SSH keys which are subject to destruction upon exercising a command. Section 3.10 of [AGD] points to the section in [CLI] where details on exercising this command are provided. |
|---|---|

#### 2.2.3.3 Tests

64 None

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

#### 2.2.4.1 TSS

65 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.

[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).

| | |
|---|---|
| **Findings:** | Encryption and decryption key sizes and modes supported by the TOE are described in section 6.2.4 of the [ST] which include 128 and 256 bit AES in CBC, CTR, and GCM mode. Associated NIST CAVP certificate numbers are present in Table 4 of section 2.3 of the [ST]. |

### 2.2.4.2 Guidance Documentation

66 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:** | Section 3.4 of [AGD] specifies that the TOE should be configured into FIPS mode. After the TOE is configured into FIPS mode, no further configuration is needed to support the claimed mode(s) and key size(s). |

### 2.2.4.3 Tests

**AES-CBC Known Answer Tests**

67 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.

68 **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.

69 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.

70 **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.

71 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.

72 **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].

73 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.

74 **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].

75 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**

76 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.

77 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**

78 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]
```

79 The ciphertext computed in the 1000$^{th}$ 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.

80 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**

81 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.

82 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.

83 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.

84 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**

85 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):

86 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.

87 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.

88 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.

89 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].

90 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**

91 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**

92 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]
```

93 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.

| **Findings:** | The vendor uses CAVP certificates C1880, C1881, and A1882 for AES in CBC, CTR and GCM modes. This is described in the Security Target in Table 4. |
|---|---|

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

### 2.2.5.1 TSS

94 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:** | Section 6.2.5 of the [ST] states support for ECDSA Signature Algorithm with key sizes of 256, 384, and 512 bits using NIST curves P-256, P-384 and P-521, respectively. |

### 2.2.5.2 Guidance Documentation

95 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:** | Section 3.4 of [AGD] specifies that the TOE should be configured into FIPS mode. After the TOE is configured into FIPS mode, no further configuration is needed to support the cryptographic algorithm and key sizes for signature services. |

### 2.2.5.3 Tests

**ECDSA Algorithm Tests**

***ECDSA FIPS 186-4 Signature Generation Test***

96 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***

97 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***

98 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.

99 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***

100 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.

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

| | |
|---|---|
| **Findings:** | The vendor uses CAVP certificates C1880, C1881, and A1882 for EC signature generation and verification. These are described in the Security Target in Table 4. |

## 2.2.6 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

### 2.2.6.1 TSS

102 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:** | Section 6.2.6 of the [ST] describes the supported hashing algorithms and their use within other TSFs. Namely, the TOE provides SHA-1, SHA-256, SHA-384 and SHA-512. These hashing algorithms are implemented in the following parts of the TSF:<br><br>a) TLS and SSH;<br>b) Published hash verification as part of trusted update validation; and<br>c) Hashing of passwords in non-volatile storage. |

### 2.2.6.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | Section 3.4 of [AGD] specifies that the TOE should be configured into FIPS mode. After the TOE is configured into FIPS mode, no further configuration is needed to support the required hash sizes. |

### 2.2.6.3 Tests

104 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.

105 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**

106 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**

107 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**

108 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**

109 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**

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

| | |
|---|---|
| **Findings:** | The vendor uses CAVP certificates C1880, C1881, and A1882 for SHA1 and SHA2 cryptographic hashing. These are described in the Security Target in Table 4. |

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

### 2.2.7.1 TSS

111 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:** | Values used by the HMAC function are summarized within Table 14 of section 6.2.7 of the [ST]. |

### 2.2.7.2 Guidance Documentation

112 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:** | Section 3.4 of [AGD] specifies that the TOE should be configured into FIPS mode. After the TOE is configured into FIPS mode, no further configuration is needed to support values used for the keyed hash. |

### 2.2.7.3 Tests

113 For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have

the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and message data using a known good implementation.

| | |
|---|---|
| **Findings:** | The vendor uses CAVP certificates C1880, C1881, and A1882 for keyed hashing leveraging SHA1 and SHA2 from the previous SFR component. These are described in the Security Target in Table 4. |

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

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

#### 2.2.8.1 TSS

115 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:** | Section 6.2.9 of the [ST] states: "The TOE contains a CTR_DRBG that is seeded from the hardware entropy source. Entropy from the noise is conditioned and used to seed the DRBG with 256 bits of full entropy." |

#### 2.2.8.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | Section 3.4 of [AGD] specifies that the TOE should be configured into FIPS mode. After the TOE is configured into FIPS mode, no further configuration is needed for the RNG to properly function. |

#### 2.2.8.3 Tests

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

118 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).

119 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.

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

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

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

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

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

| | |
|---|---|
| **Findings:** | The vendor uses CAVP certificates C1880, C1881, and A1882 for RBG operations. This is described in the Security Target in Table 4. |

# 2.3 Identification and Authentication (FIA)

## 2.3.1 FIA_AFL.1 Authentication Failure Management

### 2.3.1.1 TSS

121 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:** | This information is described in section 6.3.5 of the [ST]. Specifically, the TOE will count sequential authentication failures for a given user. When the administrator-defined maximum number of authentication failures is reached for the user account, the account is locked out until an administrator-defined time limit expires. |

122 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:** | As is stated in section 6.3.5 of the [ST], the local console does not implement the lock out mechanism. |

### 2.3.1.2 Guidance Documentation

123 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:** | The Configuring the System and Ports->Configuring Access Control->User and IP Lockout Settings section in [USER] contains instructions for configuring the number of successive unsuccessful authentication attempts and the time period. There is not a process for the remote administrator to log on with the exception of waiting for the timeout to expire. The same mechanism is used for SSH and TLS and there is no lockout mechanism for the local console. |

124 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:** | No description is necessary. Section 6.3.5 of [ST] states that the local console does not implement the lockout mechanism which ensures that administrator access is always maintained. |

#### 2.3.1.3 Tests

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

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

| **High-Level Test Description** |
|---|
| The evaluator configured the authentication failure threshold to 3 attempts and a lockout for 5 minutes. The evaluator then attempted to log in with the incorrect password 3 times via the Web GUI and confirmed that the user was locked out. The evaluator then attempted to log in from a separate IP using the correct credentials and confirmed the user was still locked out, showing the lockout was not IP based. The evaluator then waited 4 minutes 30 seconds and attempted to log in with the correct credentials and confirmed that the user was still locked out. The evaluator then waited until the timer reached 5 minutes and successfully logged in with the correct credentials. This test was then repeated via the CLI. |
| Findings: PASS |

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).

| **High-Level Test Description** |
|---|
| The TOE only claims time-based lockout. |

| High-Level Test Description |
| --- |
| Findings: PASS |

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

| High-Level Test Description |
| --- |
| This was performed in conjunction with test 1 above. |
| Findings: PASS |

### 2.3.2 FIA_PMG_EXT.1 Password Management

#### 2.3.2.1 TSS

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

| | |
| --- | --- |
| **Findings:** | Section 6.3.1 of the [ST] states: “The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper- and lower-case letters, numbers, and special characters “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”. The minimum password length is settable by the Administrator and can range from 9 to 15 characters.” |

#### 2.3.2.2 Guidance Documentation

127 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:** | Section Configuring the System and Ports->Configuring Access Control->Password Policies->Minimum Password Length and Character Requirements of [USER] specifies guidance on the composition of strong passwords and setting the minimum password length. Section 3.8 of [AGD] states the settable values for the minimum password length and the special characters that can be used in password composition. |

#### 2.3.2.3 Tests

128 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.

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

| **High-Level Test Description** |
|---|
| The evaluator changed the minimum password length to 15 characters and requiring 1 uppercase character, 1 special character, 1 lowercase character, and 1 numerical character. The evaluator then attempted to change the user's password to passwords that either complied or violated this policy. The evaluator confirmed that the password was accepted when it was expected to and it was rejected when expected. |
| Findings: PASS |

### 2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

#### 2.3.3.1 TSS

129 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:** | Section 6.3.2 of the [ST] states: "The TOE requires all users to be successfully identified and authenticated." This section also lists the supported logon methods, namely, local, remote/SSH and remote/HTTPS.<br><br>Section 6.3.3 of the [ST] states: "Regardless of the interface at which the administrator interacts, the TOE prompts the user for a credential. Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality. No TOE administrative access is permitted until an administrator is successfully identified and authenticated. ""<br><br>Section 6.3.3 of the [ST] also states: "The TOE provides a local password-based authentication mechanism.<br><br>The process for authentication is the same for administrative access whether administration is occurring via direct connection or remotely. At initial login, the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative credential associated with the user account (e.g., password or SSH public/private key response). The TOE then either grants administrative access (if the combination of username and credential is correct) or indicates that the login was unsuccessful. The TOE does not provide a reason for failure in the cases of a login failure." |

130 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:** | Section 6.3.2 of the [ST] states: "The TOE warning banner and TOE version may be viewed prior to authentication."<br><br>Furthermore, section 6.3.3 of the [ST] states: "No TOE administrative access is permitted until an administrator is successfully identified and authenticated." |

131 For distributed TOEs the evaluator shall examine that the TSS details how Security Administrators are authenticated and identified by all TOE components. If not all TOE components support authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the TSS shall describe how the overall TOE functionality is split between TOE components including how it is ensured that no unauthorized access to any TOE component can occur.

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

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

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

#### 2.3.3.2 Guidance Documentation

133 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:** | [INSTALL] specifies the steps necessary to be performed when logging into the TOE for the first time. [AGD] section 3.5 instructs the Security Administrator to change the default account's password after logging in. |

#### 2.3.3.3 Tests

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

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

| High-Level Test Description |
|---|
| The evaluator confirmed that a user could successfully log in via the Web GUI using the correct password. The evaluator then attempted to log in via the Web GUI using an incorrect password and confirmed access to the TOE was not granted.<br><br>The evaluator repeated this test for the SSH CLI and local console.. |
| Findings: PASS |

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

| High-Level Test Description |
|---|
| The evaluator confirmed that the only service available on the Web GUI before log in was viewing the log in banner. |
| Findings: PASS |

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

| High-Level Test Description |
|---|
| The evaluator confirmed that the only service available on the CLI before log in was viewing the log in banner. |
| Findings: PASS |

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

| **Findings:** | The TOE is not a distributed TOE. |
|---|---|

### 2.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

135 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.

### 2.3.5 FIA_UAU.7 Protected Authentication Feedback

#### 2.3.5.1 TSS

136 None

#### 2.3.5.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | The TOE provides obscured feedback of passwords by default; no further steps need to be taken. |

#### 2.3.5.3 Tests

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

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

| **High-Level Test Description** |
|---|
| The evaluator confirmed that when logging into the local console that there was not password feedback echoed back to the user. |
| Findings: PASS |

## 2.4 Security management (FMT)

### 2.4.1 General requirements for distributed TOEs

#### 2.4.1.1 TSS

139 For distributed TOEs it is required to verify the TSS to ensure that it describes how every function related to security management is realized for every TOE component and shared between different TOE components. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

#### 2.4.1.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

#### 2.4.1.3 Tests

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

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

### 2.4.2 FMT_MOF.1/ManualUpdate

#### 2.4.2.1 TSS

142 For distributed TOEs see chapter 2.4.1.1. There are no specific requirements for non-distributed TOEs.

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

#### 2.4.2.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | [AGD] section 2.4 specifies that [USER] section Maintenance->Upgrading PFOS contains instructions to update the TOE. This section points to ***PFOS 6.x Release Notes*** for procedures to update the TOE which state that the TOE will reboot during the install process. Additionally, section 2.2 of [AGD] instructs the Security Administrator to verify the TOE software against a published hash. |

144 For distributed TOEs the guidance documentation shall describe all steps how to update all TOE components. This shall contain description of the order in which components need to be updated if the order is relevant to the update process. The guidance documentation shall also provide warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update (if applicable).

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

#### 2.4.2.3 Tests

145 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.

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

| **High-Level Test Description** |
|---|
| The evaluator attempted to update the TOE as an unauthorized user and confirmed the attempt was rejected. Attempts to update the TOE via an authorized user were performed in conjunction with FPT_TUD_EXT.1. |
| Findings: PASS |

### 2.4.3 FMT_MTD.1/CoreData Management of TSF Data

#### 2.4.3.1 TSS

147 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:** | Section 6.4.3 of the [ST] states: "Users are required to login before being provided with access to any administrative functions. The TOE restricts the ability to manage the TSF data to Security Administrators."<br><br>Furthermore, section 6.4.4 of the [ST] states: "Management of TSF data via the CLI or web GUI is restricted to Security Administrators." |

148 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:** | Section 6.4.3 of the [ST] states: "Users are required to login before being provided with access to any administrative functions. The TOE restricts the ability to manage the TSF data to Security Administrators."<br><br>Furthermore, section 6.4.4 of the [ST] states: "Management of TSF data via the CLI or web GUI is restricted to Security Administrators." |

#### 2.4.3.2 Guidance Documentation

149 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:** | [USER] section Configuring the System and Ports->Configuring Access Control contains instructions for creating users and applying the Security Administrator role. Note that only Security Administrators have the ability to manage TSF data. |

150 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:** | Section 3.4 of [AGD] provides instructions on how to install certificates and how to designate certificates as trusted CA certificates. |

#### 2.4.3.3 Tests

151 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.

### 2.4.4 FMT_SMF.1 Specification of Management Functions

152 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.

#### 2.4.4.1 TSS (containing also requirements on Guidance Documentation and Tests)

153 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:** Section 6.4.6 of the [ST] states: "The TOE may be managed via the CLI (console & SSH) or GUI (HTTPS). The specific management capabilities include:

a) Ability to administer the TOE locally and remotely
b) Ability to configure the access banner
c) Ability to configure the session inactivity time before session termination or locking
d) Ability to update the TOE and to verify the updates
e) Ability to configure the authentication failure parameters
f) Ability to configure audit behavior (enable/disable remote logging)
g) Ability to set the time which is used for timestamps
h) Ability to manage the cryptographic keys, including import and management of X.509v3 certificates
i) Ability to manage the trusted public keys database."

The guidance activities are covered by the activities for the following SFRs:

a) FMT_SMR.2
b) FTS_TAB.1
c) FTA_SSL_EXT.1 and FTA_SSL.3
d) FPT_TUD_EXT.1
e) FIA_AFL.1
f) FMT_MOF.1/Functions
g) FTA_STM_EXT.1
h) FMT_MTD.1/CryptoKeys
i) FMT_MTD.1/CryptoKeys

154 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:** Section 6.4.6 of the [ST] describes three administrative interfaces: a local CLI interface, a remote CLI interface (via SSH) and a GUI (via HTTPS).

[AGD] section 3.2 describes the local and remote administrative interfaces. The local interface is physically separated from the remote interfaces which is distinctly obvious to the administrator that it is local.

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

**Findings:** The TOE is not a distributed TOE.

#### 2.4.4.2 Guidance Documentation

156 See section 2.4.4.1.

#### 2.4.4.3 Tests

157 The evaluator tests management functions as part of testing the SFRs identified in section 4.4.4. No separate testing for FMT_SMF.1 is required unless one of the management functions in FMT_SMF.1.1 has not already been exercised under any other SFR.

### 2.4.5 FMT_SMR.2 Restrictions on security roles

#### 2.4.5.1 TSS

158 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:** | Section 6.4.4 of the [ST] describes a single role of Security Administrator. Management of TSF data via the local CLI, remote CLI, or web-based GUI is restricted to Security Administrators. |

#### 2.4.5.2 Guidance Documentation

159 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:** | [INSTALL] provides instructions for how to administer to the TOE via the Web UI, SSH CLI and Local CLI. |

#### 2.4.5.3 Tests

160 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.

| **High-Level Test Description** |
|---|
| The evaluator tested all interfaces through testing of the other test activities. |
| Findings: PASS |

## 2.5 Protection of the TSF (FPT)

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

#### 2.5.1.1 TSS

161 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:** | Table 15 of section 6.5.1 of the [ST] describes various keys and how they are stored on the device. All keys are stored in plaintext. The section specifically states: “In all cases, plaintext keys cannot be viewed through an interface designed specifically for that purpose.” |

### 2.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

#### 2.5.2.1 TSS

162 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:** | Table 16 of section 6.5.2 of the [ST] describes the generation, obfuscation, storage and zeroization of locally stored administrator passwords. According to the table, plaintext passwords are hashed using SHA-256 prior to storage. The section also states: “In all cases plaintext passwords cannot be viewed through an interface designed specifically for that purpose.” |

### 2.5.3 FPT_TST_EXT.1 TSF testing

#### 2.5.3.1 TSS

163 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:** | Section 6.5.3 of the [ST] describes memory and HW component tests present in the POST. This section also provides a detailed description of a cryptographic POST which consists of a software integrity test, KATs, PCTs, and random bit/number generation tests. |

164 For distributed TOEs the evaluator shall examine the TSS to ensure that it details which TOE component performs which self-tests and when these self-tests are run.

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

#### 2.5.3.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | [AGD] section 2.3 lists the POST and FIPS Self-tests that are run by the TOE and possible errors that may occur. This section also provides actions for the Security |

| | |
|---|---|
| | Administrator to take in the event that the TOE encounters an error. These tests align with the tests described in section 6.5.3 of [ST] |

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

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

#### 2.5.3.3 Tests

167 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.

168 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.

169 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.

170 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.

| **High-Level Test Description** |
|---|
| The evaluator reset the TOE and observed that the system integrity self-tests were executed via the audit log. The evaluator also observed the feedback from the local console during the TOE reset and observed that the TOE properly initialized. |
| Findings: PASS |

### 2.5.4 FPT_TUD_EXT.1 Trusted Update

#### 2.5.4.1 TSS

171 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:** | Section 6.5.4 of the [ST] states: “The current firmware version may be queried using either the CLI or the Web GUI.” Section 6.5.4 of the [ST] also states that only administrators may initiate updates to the TOE. |

172 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:** | Section 6.5.4 of the [ST] states: "The administrator may download a firmware update from the trusted source and verify it using the published hash. Namely, an administrator must download the update along with the SHA1 checksum associated with the file from the NETSCOUT support website to their local machine.<br><br>Only administrators may initiate updates to the TOE. The administrator should discard unsuccessfully validated images, otherwise the update should be applied to the TOE." |

173 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:** | These selections are not made in the FPT_TUD_EXT.1.2 requirement of the [ST]. |

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

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

175 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:** | Section 6.5.4 of the [ST] states: "The administrator may download a firmware update from the trusted source and verify it using the published hash." And "Only administrators may initiate updates to the TOE." Verification of the published hash is a manual process. |

### 2.5.4.2 Guidance Documentation

176 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:** | Section 2.2 of the [AGD] points to section Managing with PFOS->Using the Web UI->System Status in [USER] which includes the Software Tab to query the currently active version. Section 2.4 of the [AGD] also addresses the TOE's support of delayed activation where the process to query the 'current' and 'nextboot' versions are found in [REL] Section 3. |

177 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.

178 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:** | Section 2.2 of [AGD] specifies that the TOE is verified by a SHA256 published hash that is provided by the developer's support website which is consistent with section 6.5.4 of the [ST]. Successful validation results in the image being applied and unsuccessful attempts should result in the image being discarded. |

179 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:** | The TOE is not a distributed TOE. |

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

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

181 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:** | The TOE update is not verified by using a digital signature. |

#### 2.5.4.3 Tests

182 The evaluator shall perform the following tests:

a. Test 1: The evaluator performs the version verification activity to determine the current version of the product. If a trusted update can be installed on the TOE with a delayed activation, the evaluator shall also query the most recently installed version (for this test the TOE shall be in a state where these two versions

match). The evaluator obtains a legitimate update using procedures described in the guidance documentation and verifies that it is successfully installed on the TOE. For some TOEs loading the update onto the TOE and activation of the update are separate steps ('activation' could be performed e.g. by a distinct activation step or by rebooting the device). In that case the evaluator verifies after loading the update onto the TOE but before activation of the update that the current version of the product did not change but the most recently installed version has changed to the new product version. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again.

| **High-Level Test Description** |
| --- |
| The evaluator queried the version of the TOE then calculated the SHA256 hash of the update file and confirmed it matched the published value. The evaluator then applied the update to the TOE and queried the version again to confirm the update was successful. |
| Findings: PASS |

b. Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator 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:** | The TOE update is not verified by using a digital signature. |
| --- | --- |

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.

183 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:** | The TOE does not support published hashes. All published hashes are compared by the Security Administrator offline. |

184 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).

| | |
|---|---|
| **Findings:** | The TOE only supports manual updates. The test cases above are not applicable to automatic checking of updates since there are no images to install during an automatic check. |

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

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

## 2.5.5 FPT_STM_EXT.1 Reliable Time Stamps

### 2.5.5.1 TSS

186 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:** | Section 6.5.5 of the [ST] states the following:<br><br>"The TOE incorporates an internal clock that is used to maintain date and time. The Security Administrator sets the date and time during initial TOE configuration and may change the time during operation.<br><br>The TOE makes used of time for the following:<br><br>a) Audit record timestamps<br>b) Session timeouts (lockout enforcement)<br>c) Certificate validation" |

187 [Modified by TD0632] If "obtain time from the underlying virtualization system" is selected, the evaluator shall examine the TSS to ensure that it identifies the VS interface the TOE uses to obtain time. If there is a delay between updates to the time on the VS and updating the time on the TOE, the TSS shall identify the maximum possible delay.

| | |
|---|---|
| **Findings:** | This is not applicable because the selection is not made by [ST]. |

### 2.5.5.2 Guidance Documentation

188 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:** | Section 3.6 of [AGD] specifies how the TOE's time can be set manually. The TOE does not support NTP. |
|---|---|

189 [Modified by TD0632] If the TOE supports obtaining time from the underlying VS, the evaluator shall verify the Guidance Documentation specifies any configuration steps necessary. If no configuration is necessary, no statement is necessary in the Guidance Documentation. If there is a delay between updates to the time on the VS and updating the time on the TOE, the evaluator shall ensure the Guidance Documentation informs the administrator of the maximum possible delay.

| **Findings:** | This is not applicable because the TOE is not a VM and does not receive time from an underlying VS. |
|---|---|

### 2.5.5.3 Tests

190 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.

| **High-Level Test Description** |
|---|
| The evaluator queried the time on the TOE then attempted to change the time. The evaluator queried the time on the TOE again and confirmed that the time had successfully changed. |
| Findings: PASS |

b. Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE, and set up a communication path with the NTP server. The evaluator 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:** | The ST does not claim that the TOE supports the use of NTP. |
|---|---|

c. [Modified by TD0632] 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:** | This is not applicable because the TOE is not a VM and does not receive time from an underlying VS. |
|---|---|

191 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:** | The TOE does not support independent time information. |
|---|---|

## 2.6 TOE Access (FTA)

### 2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

#### 2.6.1.1 TSS

192 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:** | Section 6.6.1 of the [ST] states: “The Security Administrator may configure the TOE to terminate an inactive local interactive session (CLI) following a specified period. The timeout value is set to thirty minutes by default.” |
|---|---|

#### 2.6.1.2 Guidance Documentation

193 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:** | Section 3.3 of [AGD] specifies how the session termination value can be set for the local CLI/Console. |
|---|---|

#### 2.6.1.3 Tests

194 The evaluator shall perform the following test:

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

| **High-Level Test Description** |
|---|
| The evaluator set the timeout value to 5 minutes then logged in to the CLI. The evaluator waited 5 minutes and confirmed that the user was successfully logged out. The evaluator then repeated this test with a 8 minute inactivity threshold. |
| Findings: PASS |

### 2.6.2 FTA_SSL.3 TSF-initiated Termination

#### 2.6.2.1 TSS

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

| | |
|---|---|
| **Findings:** | Section 6.6.2 of the [ST] states: "The Security Administrator may configure the TOE to terminate an inactive remote interactive session (CLI and Web UI) following a specified period. The timeout value is set to thirty minutes by default." |

#### 2.6.2.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | Section 3.3 of [AGD] specifies how the session termination value can be set for remote CLI/SSH and GUI/HTTPS access. |

#### 2.6.2.3 Tests

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

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

| **High-Level Test Description** |
|---|
| The evaluator set the timeout value to 5 minutes then logged in to the Web GUI and SSH CLI. The evaluator waited 5 minutes and confirmed that the user was successfully logged out. The evaluator then repeated this test with a 8 minute inactivity threshold. |
| Findings: PASS |

### 2.6.3 FTA_SSL.4 User-initiated Termination

#### 2.6.3.1 TSS

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

| | |
|---|---|
| **Findings:** | Section 6.6.3 of the [ST] states: "Administrative users may terminate their own sessions by logging out." |

#### 2.6.3.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | Section 3.3 of the [AGD] covers the procedure for terminating both local and remote interactive sessions by typing a command in the CLI or using the 'Logout' button on the Web UI. |

#### 2.6.3.3 Tests

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

a. Test 1: The evaluator initiates an interactive local session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated.

| **High-Level Test Description** |
|---|
| The evaluator logged into the TOE via the CLI and typed 'exit' to quit the session and confirmed that the user was successfully logged out. |
| Findings: PASS |

b. Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated.

| **High-Level Test Description** |
|---|
| The evaluator logged in to the TOE via the Web GUI and clicked the logout button and confirmed that the user was successfully logged out. The evaluator then logged into the TOE via the SSH CLI and typed 'exit' to quit the session and confirmed that the user was successfully logged out. |
| Findings: PASS |

### 2.6.4 FTA_TAB.1 Default TOE Access Banners

#### 2.6.4.1 TSS

201 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:** | Section 6.6.4 of the [ST] states: "The TOE displays an administrator configurable message to users prior to login at the local CLI, the remote CLI, and web GUI." |

#### 2.6.4.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | [AGD] Section 3.3 describes how the banner message can be configured for both local and remote interfaces. |

#### 2.6.4.3 Tests

203 The evaluator shall also perform the following test:

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

| **High-Level Test Description** |
|---|
| The evaluator logged into the TOE and changed the log in banner. The evaluator then logged out and attempted to log in via the Web GUI, local console and SSH CLI and confirmed the banner was displayed before logging in. |
| Findings: PASS |

## 2.7 Trusted path/channels (FTP)

### 2.7.1 FTP_ITC.1 Inter-TSF trusted channel

#### 2.7.1.1 TSS

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

| | |
|---|---|
| **Findings:** | Section 6.7.1 of the [ST] describes the usage of SSH to communicate with an external audit (Syslog) server in conformance with FCS_SSHC_EXT.1. |

#### 2.7.1.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | [AGD] section 3.7 specifies how a connection to the SSH syslog server can be established and lists the configuration options in case the connection is broken. No other connections/entities are claimed. |

#### 2.7.1.3 Tests

206 The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no

expectation that this information must be recorded in any public-facing document or report.

207 The evaluator shall perform the following tests:

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

| **High-Level Test Description** |
| --- |
| The only trusted channel is the remote audit log to an SSH server, which is set up as per the evaluated configuration. It is constantly tested throughout the evaluation. |
| Findings: PASS |

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

| **High-Level Test Description** |
| --- |
| The ST only claims SSH for audit log transfer. This test is performed in conjunction with FCS_SSHC_EXT testing. |
| Findings: PASS |

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

| **High-Level Test Description** |
| --- |
| The ST only claims SSH for audit log transfer. This test is performed in conjunction with FCS_SSHC_EXT testing. |
| Findings: PASS |

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

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

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

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

| **High-Level Test Description** |
|---|
| The evaluator performed 2 tests, one short disconnect and one long disconnect. The evaluator logged into the TOE and generated audit messages to be sent to the syslog server. The evaluator confirmed communications to each server were encrypted then physically disconnected the server from an intermediate switch. The evaluator then attempted to log in and generate audit messages and confirmed the TOE attempted to communicate with the server but was unsuccessful and the channel data was not sent in plaintext. The evaluator then reconnected the servers and attempted to log in and generate audit messages again and confirmed that the communications were restored and the channel data was sent encrypted. |
| Findings: PASS |

Further assurance activities are associated with the specific protocols.

208 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:** | This is not a distributed TOE. |

209 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.

## 2.7.2 FTP_TRP.1/Admin Trusted Path

### 2.7.2.1 TSS

210 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:** | Section 6.7.2 of the [ST] states:<br><br>“The TOE provides the following trusted paths for remote administration:<br><br>a) Web GUI over HTTPS per FCS_HTTPS_EXT.1.1<br>b) CLI over SSH per FCS_SSHS_EXT.1.1”<br><br>The evaluator confirmed these protocols are present within the FTP_TRP.1/Admin requirement of section 5.3 of the [ST]. Furthermore, the evaluator confirmed the FCS_HTTPS_EXT.1, FCS_SSHS_EXT.1, and FCS_TLSS_EXT.1 components in |

| | |
|---|---|
| | section 5.3.2 of the [ST] are consistent with the trusted paths for remote administration. |

#### 2.7.2.2 Guidance Documentation

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

| | |
|---|---|
| **Findings:** | [INSTALL] specifies how an administrator can establish an SSH or HTTPS session to connect to the TOE for remote administration. |

#### 2.7.2.3 Tests

212 The evaluator shall perform the following tests:

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

| **High-Level Test Description** |
|---|
| The only trusted paths are the web interface (HTTPS) and the SSH CLI, which are both set up as per the evaluated configuration. They are constantly tested throughout the evaluation. |
| Findings: PASS |

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

| **High-Level Test Description** |
|---|
| This test is performed in conjunction with FCS_TLSS_EXT and FCS_SSHS_EXT testing. |
| Findings: PASS |

213 Further assurance activities are associated with the specific protocols.

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

| | |
|---|---|
| **Findings:** | This is not a distributed TOE. |

# 3 Evaluation Activities for Optional Requirements

215 No optional requirements have been selected by this evaluation.

# 4 Evaluation Activities for Selection-Based Requirements

## 4.1 Security Audit (FAU)

## 4.2 Cryptographic Support (FCS)

### 4.2.1 FCS_HTTPS_EXT.1 HTTPS Protocol

#### 4.2.1.1 TSS

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

| | |
|---|---|
| **Findings:** | Section 6.2.8 of the [ST] outlines how the HTTPS implementation complies with RFC 2818. Namely, section 6.2.8 of the [ST] states:<br><br>"The TOE's HTTPS protocol complies with RFC 2818.<br><br>RFC 2818 specifies HTTP over TLS. The majority of RFC 2818 is spent on discussing practices for validating endpoint identities and how connections must be setup and torn down. The TOE web GUI operates on an explicit port designed to natively speak TLS: it does not attempt STARTTLS or similar multi-protocol negotiation which is described in section 2.3 of RFC 2818. The web server uses a variant of OpenSSL which attempts to send closure Alerts prior to closing a connection in accordance with section 2.2.2 of RFC 2818." |

#### 4.2.1.2 Guidance Documentation

217 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:** | The TOE does not act as an HTTPS client. [INSTALL] provides instructions for how the Administrator can successfully connect to the Web GUI HTTPS server. |

#### 4.2.1.3 Tests

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

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

220 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.

### 4.2.2 FCS_SSHC_EXT.1 SSH Client

#### 4.2.2.1 TSS

**FCS_SSHC_EXT.1.2**

221 [Modified by TD0636] The evaluator shall check to ensure that the TSS contains a list of the public key algorithms that are acceptable for use for user authentication and

that this list is consistent with asymmetric key generation algorithms selected in FCS_CKM.1, hashing algorithms selected in FCS_COP.1/Hash, and signature generation algorithms selected in FCS_COP.1/SigGen. The evaluator shall confirm the TSS is unambiguous in declaring the TOE's ability to authenticate itself to a remote endpoint with a user-based public key.

222 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:** | Section 6.2.10 of the [ST] states: "The TOE supports public key authentication (ecdsa-sha2-nistp256)." This is consistent with the selections made in the FCS_CKM.1, FCS_COP.1/Hash, FCS_COP.1/SigGen and FCS_SSHC_EXT.1.5 requirements of the [ST]. |

**FCS_SSHC_EXT.1.3**

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

| | |
|---|---|
| **Findings:** | Section 6.2.10 of the [ST] states: "The TOE examines the size of each received SSH packet. If the packet is greater than 256 KB, it is automatically dropped." |

**FCS_SSHC_EXT.1.4**

224 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:** | Section 6.2.10 of the [ST] states: "The TOE utilises AES-CTR-128, AES-CTR-256, AES-GCM-128 and AES-GCM-256 for SSH encryption." These are consistent with the selections made in the FCS_SSHC_EXT.1.4 requirement of the [ST]. |

**FCS_SSHC_EXT.1.5**

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

226 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:** | Section 6.2.10 of the [ST] states: "The TOE utilised ECDSA-SHA2-NISTP256, for its public key algorithm when using public key authentication." This is consistent with the selections made in the FCS_SSHC_EXT.1.5 requirement of the [ST]. |

227 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:** | x509v3-based public key authentication is not used by the TOE to establish the server's identity. |

**FCS_SSHC_EXT.1.6**

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

| | |
|---|---|
| **Findings:** | Section 6.2.10 of the [ST] states: "The TOE provides data integrity for SSH connections via HMAC-SHA1, HMAC-SHA2-256 and HMAC-SHA2-512." This is consistent with the selections made in the FCS_SSHC_EXT.1.6 requirement of the [ST]. |

**FCS_SSHC_EXT.1.7**

229 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:** | Section 6.2.10 of the [ST] states: "The TOE supports ecdh-sha2-nistp256, ecdh-sha2-nistp384 and ecdh-sha2-nistp521, for SSH key exchanges." This is consistent with the selections made in the FCS_SSHC_EXT.1.7 requirement of the [ST]. |

**FCS_SSHC_EXT.1.8**

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

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

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

#### 4.2.2.2 Guidance Documentation

**[Modified by TD0636] FCS_SSHC_EXT.1.2**

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

| | |
|---|---|
| **Findings:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the algorithms claimed in the requirements. |

**FCS_SSHC_EXT.1.4**

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

| | |
|---|---|
| **Findings:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the algorithms claimed in the requirements. |

**FCS_SSHC_EXT.1.5**

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

| | |
|---|---|
| **Findings:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the algorithms claimed in the requirements. |

**FCS_SSHC_EXT.1.6**

234 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:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the algorithms claimed in the requirements. |

**FCS_SSHC_EXT.1.7**

235 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:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the algorithms claimed in the requirements. |

**FCS_SSHC_EXT.1.8**

236 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:** | The TOE rekeys at 1 hour or after 1 gigabyte of data is passed. These values are static and not configurable on the TOE. |

#### 4.2.2.3 Tests

**[Modified by TD0636] FCS_SSHC_EXT.1.2**

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

238 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:** | The TOE only supports ecdsa-sha2-nistp256 keys for user public key authentication. This test was performed in conjunction with FCS_SSHC_EXT.1.5. |

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:** | This test is not applicable because the TOE does not support password-based authentication. |
|---|---|

**FCS_SSHC_EXT.1.3**

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

| **High-Level Test Description** |
|---|
| The evaluator established an SSH connection with the TOE and sent a packet just under the threshold and confirmed it was accepted. The evaluator then established an SSH connection to the TOE and sent a packet just over the threshold and confirmed it was rejected by the TOE. |
| Findings: PASS |

**FCS_SSHC_EXT.1.4**

240 The evaluator must ensure that only claimed ciphers and cryptographic primitives are used to establish a SSH connection. To verify this, the evaluator shall start session establishment for a SSH connection with a remote server (referred to as 'remote endpoint' below). The evaluator shall capture the traffic exchanged between the TOE and the remote endpoint during protocol negotiation (e.g. using a packet capture tool or information provided by the endpoint, respectively). The evaluator shall verify from the captured traffic that the TOE offers all the ciphers defined in the TSS for the TOE for SSH sessions, but no additional ones compared to the definition in the TSS. The evaluator shall perform one successful negotiation of an SSH session to verify that the TOE behaves as expected. It is sufficient to observe the successful negotiation of the session to satisfy the intent of the test. If the evaluator detects that not all ciphers defined in the TSS for SSH are supported by the TOE and/or the TOE supports one or more additional ciphers not defined in the TSS for SSH, the test shall be regarded as failed.

| **High-Level Test Description** |
|---|
| The evaluator attempted an SSH connection with the TOE using a valid supported cipher algorithm. The evaluator confirmed that this connection succeeded, and that the algorithm was used. The evaluator then analysed the TOE's advertised cipher algorithms and confirmed they are consistent with the claims in the Security Target. |
| Findings: PASS |

**FCS_SSHC_EXT.1.5**

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using valid specified host and user keys. The evaluator confirmed that the keys were accepted and the connection succeeded. |
| Findings: PASS |

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using an invalid host key. The evaluator confirmed that the key was rejected by the TOE and the connection failed. |
| Findings: PASS |

**FCS_SSHC_EXT.1.6**

243 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.

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using each HMAC listed in the Security Target. The evaluator confirmed that each HMAC was accepted and a connection was established. |
| Findings: PASS |

245 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.

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using hmac-md5. The evaluator confirmed that the connection was denied by the TOE. |
| Findings: PASS |

**FCS_SSHC_EXT.1.7**

247 Test 1: The evaluator shall configure an SSH server to permit all allowed key exchange methods. The evaluator shall attempt to connect from the TOE to the SSH

server using each allowed key exchange method, and observe that each attempt succeeds.

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using each key exchange algorithm listed in the Security Target. The evaluator confirmed that each key exchange algorithm was accepted and a connection was established. |
| Findings: PASS |

**FCS_SSHC_EXT.1.8**

248 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.

249 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).

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

| High-Level Test Description |
| --- |
| The evaluator established an SSH connection with the TOE and observed the connection until the time-based threshold was reached. The evaluator confirmed that upon reaching the threshold that the TOE initiated a rekey. |
| Findings: PASS |

251 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).

252 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).

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

| High-Level Test Description |
| --- |
| The evaluator established an SSH connection with the TOE and observed the connection until the data-based threshold was reached. The evaluator confirmed that upon reaching the threshold that the TOE initiated a rekey. |

| High-Level Test Description |
| --- |
| Findings: PASS |

254 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).

| **Findings:** | The thresholds are not configurable on the TOE. |
| --- | --- |

255 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

| **Findings:** | The TOE does not have hardware limitations. |
| --- | --- |

**FCS_SSHC_EXT.1.9**

256 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.

| High-Level Test Description |
| --- |
| The evaluator cleared the known host key database on the TOE. The evaluator then attempted an SSH connection with the TOE and the SSH server and confirmed upon receipt of the unknown key, the TOE rejected the connection. |
| Findings: PASS |

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

to the SSH server using public key-based authentication, and shall ensure that the TOE rejects the connection.

| High-Level Test Description |
| --- |
| The evaluator generated an unknown and non-supported host key to be used by the remote SSH server. The evaluator then attempted an SSH connection with the TOE and the SSH server and confirmed upon receipt of the unknown key, the TOE rejected the connection. |
| Findings: PASS |

### 4.2.3 FCS_SSHS_EXT.1 SSH Server

#### 4.2.3.1 TSS

**FCS_SSHS_EXT.1.2**

258 [Modified by TD0631] 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).

259 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.

260 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:** | Section 6.2.11 of the [ST] states: "The TOE supports password-based or public key authentication (ecdsa-sha2-nistp256). In the case of public keys, the TOE authenticates the identity of the SSH client using a local database associating authorized hosts with its corresponding public key." This is consistent with the selections made in the FCS_COP.1/SigGen and FCS_SSHS_EXT.1.5 requirements of the [ST]. |

**FCS_SSHS_EXT.1.3**

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

| | |
| --- | --- |
| **Findings:** | Section 6.2.11 of the [ST] states: "The TOE examines the size of each received SSH packet. If the packet is greater than 256 KB, it is automatically dropped." |

**FCS_SSHS_EXT.1.4**

262 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:** | Section 6.2.11 of the [ST] states: "The TOE utilises AES-CTR-128, AES-CTR-256, AES-GCM-128 and AES-GCM-256 for SSH encryption." This is consistent with the selections made in the FCS_SSHS_EXT.1.4 requirement of the [ST]. |

**FCS_SSHS_EXT.1.5**

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

| | |
|---|---|
| **Findings:** | Section 6.2.11 of the [ST] states: "The TOE utilised ECDSA-SHA2-NISTP256 for its public key algorithm when using public key authentication." This is consistent with the selections made in the FCS_SSHS_EXT.1.5 requirement of the [ST]. |

264 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:** | Section 6.2.11 of the [ST] states: "In the case of public keys, the TOE authenticates the identity of the SSH client using a local database associating authorized hosts with its corresponding public key." |

**FCS_SSHS_EXT.1.6**

265 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:** | Section 6.2.11 of the [ST] states: "The TOE provides data integrity for SSH connections via HMAC-SHA1, HMAC-SHA2-256 and HMAC-SHA2-512." This is consistent with the selections made in FCS_SSHS_EXT.1.6 requirement of the [ST]. |

**FCS_SSHS_EXT.1.7**

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

| | |
|---|---|
| **Findings:** | Section 6.2.11 of the [ST] states: "The TOE supports ecdh-sha2-nistp256, ecdh-sha2-nistp384 and ecdh-sha2-nistp521 for SSH key exchanges." This is consistent with the selections made in FCS_SSHS_EXT.1.7 requirement of the [ST]. |

**FCS_SSHS_EXT.1.8**

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

1. Both thresholds are checked by the TOE.

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

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

### 4.2.3.2 Guidance Documentation

**FCS_SSHS_EXT.1.4**

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

| | |
|---|---|
| **Findings:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the algorithms claimed in the requirements. |

**FCS_SSHS_EXT.1.5**

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

| | |
|---|---|
| **Findings:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the algorithms claimed in the requirements. |

**FCS_SSHS_EXT.1.6**

270 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:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the algorithms claimed in the requirements. |

**FCS_SSHS_EXT.1.7**

271 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:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the algorithms claimed in the requirements. |

**FCS_SSHS_EXT.1.8**

272 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:** | The TOE rekeys at 1 hour or after 1 gigabyte of data is passed. These values are static and not configurable on the TOE. |

#### 4.2.3.3 Tests

**[Modified by TD0631] FCS_SSHS_EXT.1.2**

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

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using a valid specified ecdsa-sha2-nistp256 key. The evaluator confirmed that the key was accepted and the connection succeeded. Note no other public key algorithms are supported by the TOE. |
| Findings: PASS |

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using an invalid key. The evaluator confirmed that the key was rejected by the TOE and the connection failed. |
| Findings: PASS |

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection to the TOE using a known good username and password and confirmed that the connection succeeded. |
| Findings: PASS |

277 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.

278

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection to the TOE using a known good username and bad password and confirmed that the connection was rejected by the TOE. |
| Findings: PASS |

**FCS_SSHS_EXT.1.3**

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

| High-Level Test Description |
| --- |
| The evaluator established an SSH connection with the TOE and sent a packet just under the threshold and confirmed it was accepted. The evaluator then established an SSH connection to the TOE and sent a packet just over the threshold and confirmed it was rejected by the TOE. |

| High-Level Test Description |
| --- |
| Findings: PASS |

**FCS_SSHS_EXT.1.4**

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using a valid supported cipher algorithm. The evaluator confirmed that this connection succeeded, and that the algorithm was used. The evaluator then analysed the TOE's advertised cipher algorithms and confirmed they are consistent with the claims in the Security Target. |
| Findings: PASS |

**[Modified by TD0631] FCS_SSHS_EXT.1.5**

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

282 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.

283 Has effectively been moved to FCS_SSHS_EXT.1.2.

| | |
| --- | --- |
| **Findings:** | The TOE only supports ecdsa-sha2-nistp256 keys for user and host public key authentication. This test was performed in conjunction with FCS_SSHS_EXT.1.2. |

284 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.

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using an unsupported key (because the public key algorithm is unsupported). The evaluator confirmed that the key was rejected by the TOE and the connection failed. |
| Findings: PASS |

**FCS_SSHS_EXT.1.6**

286 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.

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using each HMAC listed in the Security Target. The evaluator confirmed that each HMAC was accepted and a connection was established. |
| Findings: PASS |

288 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.

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using hmac-md5. The evaluator confirmed that the connection was denied by the TOE. |
| Findings: PASS |

**FCS_SSHS_EXT.1.7**

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

| High-Level Test Description |
| --- |
| The evaluator attempted an SSH connection with the TOE using the diffie-hellman-group1-sha1 key exchange algorithm. The evaluator confirmed that the TOE rejected the key exchange algorithm and a connection was not established. |
| Findings: PASS |

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

| **High-Level Test Description** |
| --- |
| The evaluator attempted an SSH connection with the TOE using each key exchange algorithm listed in the Security Target. The evaluator confirmed that each key exchange algorithm was accepted and a connection was established. |
| Findings: PASS |

**FCS_SSHS_EXT.1.8**

292 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.

293 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).

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

| **High-Level Test Description** |
| --- |
| The evaluator established an SSH connection with the TOE and observed the connection until the time-based threshold was reached. The evaluator confirmed that upon reaching the threshold that the TOE initiated a rekey. |
| Findings: PASS |

295 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).

296 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).

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

| **High-Level Test Description** |
| --- |
| The evaluator established an SSH connection with the TOE and observed the connection until the data-based threshold was reached. The evaluator confirmed that upon reaching the threshold that the TOE initiated a rekey. |

| High-Level Test Description |
| --- |
| Findings: PASS |

298 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).

| **Findings:** | The TOE does not support configurable thresholds. |
| --- | --- |

299 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.

| **Findings:** | The TOE does not have hardware limitations. |
| --- | --- |

## 4.2.4 FCS_TLSS_EXT.1 Extended: TLS Server Protocol Without Mutual Authentication

### 4.2.4.1 TSS

**FCS_TLSS_EXT.1.1**

300 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:** | Section 6.2.12 of the [ST] states:<br><br>"The server only allows TLS protocol versions 1.2 (rejecting any other protocol version, including SSL 2.0, SSL 3.0 and TLS 1.0 and any other unknown TLS version string supplied) and is restricted to the following ciphersuites by default:<br><br>a) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492<br>b) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492<br>c) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289<br>d) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289<br>e) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289<br>f) TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289" |
| --- | --- |

| | |
|---|---|
| | This is consistent with the selections made in the FCS_TLSS_EXT.1.1 requirement of the [ST]. |

**FCS_TLSS_EXT.1.2**

301 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:** | Section 6.2.12 of the [ST] states: "The server only allows TLS protocol versions 1.2 (rejecting any other protocol version, including SSL 2.0, SSL 3.0 and TLS 1.0, TLS 1.1 and any other unknown TLS version string supplied) …"<br><br>This is consistent with what is stated in the FCS_TLSS_EXT.1.1 and FCS_TLSS_EXT.1.2 requirements in section 5.3.2 of the [ST] which state: "The TSF shall implement [TLS 1.2 (RFC 5246)] and reject all other TLS and SSL versions." and "The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [TLS 1.1]." |

**FCS_TLSS_EXT.1.3**

302 [Modified by TD0635] 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:** | Section 6.2.12 of the [ST] states: "The TLS server can negotiate ciphersuites that include ECDHE key agreement schemes." This section also indirectly provides key agreement parameter descriptions by referencing RFC 4492 and RFC 5289. |

**FCS_TLSS_EXT.1.4**

303 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).

| | |
|---|---|
| **Findings:** | Section 6.2.12 of the [ST] states: "The TOE supports session resumption based on session tickets. -The TOE supports session tickets according to RFC5077" |

304 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.

| | |
|---|---|
| **Findings:** | A description of session ticket handling/encryption is indirectly provided in section 6.2.12 of the [ST] by reference to RFC 5077. |

305 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:** | A description of session ticket structural format is indirectly provided in section 6.2.12 of the [ST] by reference to RFC 5077. |

306 [Modified by TD0569] 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:** | A description of session resumption is indirectly provided in section 6.2.12 of the [ST] by reference to RFC 5077. |

#### 4.2.4.2 Guidance Documentation

**FCS_TLSS_EXT.1.1**

307 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:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the ciphersuites claimed in the requirements. |

**FCS_TLSS_EXT.1.2**

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

| | |
|---|---|
| **Findings:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will reject all SSL and TLS versions specified in the requirement. |

**FCS_TLSS_EXT.1.3**

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

| | |
|---|---|
| **Findings:** | [AGD] section 3.4 instructs the user to enable FIPS mode on the TOE. Once configured, the TOE will only support the key establishment curves specified claimed in the requirement. |

**[Modified by TD0569] FCS_TLSS_EXT.1.4**

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

| | |
|---|---|
| **Findings:** | The TOE does not need any additional configuration to meet the requirement. |

#### 4.2.4.3 Tests

**FCS_TLSS_EXT.1.1**

311 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).

| High-Level Test Description |
| --- |
| The evaluator successfully attempted TLS connections to the TOE using each of the claimed ciphersuites in the [ST]. |
| Findings: PASS |

312 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.

| High-Level Test Description |
| --- |
| The evaluator attempted a TLS connection to the TOE using TLS_RSA_WITH_RC4_128_SHA and verified that the connection was rejected. The evaluator then attempted another connection to the TOE using TLS_NULL_WITH_NULL_NULL and verified that it was also rejected. |
| Findings: PASS |

313 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.

| High-Level Test Description |
| --- |
| The evaluator attempted a connection to the TOE with a modified byte in the Client Finished message and verified that the connection was rejected. |
| Findings: PASS |

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'.

| **High-Level Test Description** |
|---|
| The evaluator successfully attempted a TLS connection to the TOE and verified via a packet capture that the Server Finished message was appropriately encrypted. |
| Findings: PASS |

**FCS_TLSS_EXT.1.2**

314 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.

| **High-Level Test Description** |
|---|
| The evaluator attempted independent connections to the TOE using each of the protocol versions selected in the [ST] as protocols to be denied and verified that the connection attempts were rejected. |
| Findings: PASS |

**FCS_TLSS_EXT.1.3**

315 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.

| **High-Level Test Description** |
|---|
| The evaluator successfully attempted TLS connections to the TOE using each of the claimed key exchanges in the [ST]. |
| Findings: PASS |

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.

| **High-Level Test Description** |
|---|
| The evaluator attempted a connection to the TOE using the secp192r1 curve and verified that the TOE rejected the connection. |
| Findings: PASS |

316 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:** | This test is not applicable because the TOE does not support any DHE ciphersuites. |
|---|---|

317 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:** | This test is not applicable because the TOE does not support any RSA key establishment ciphersuites. |
|---|---|

**FCS_TLSS_EXT.1.4**

318 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).

319 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.

320 [Modified by TD0569] 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.

| **Findings:** | This test is not applicable because the TOE supports session tickets. |
|---|---|

321 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).

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.

322 [Modified by TD0569] 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.

| **Findings:** | This test is not applicable because the TOE does not support session resumption using session IDs. |
|---|---|

323 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):

a) [Modified by TD0556] 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.

b) [Not modified as per TD0555] 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.

324 [Modified by TD0569] 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.

| **High-Level Test Description** |
|---|
| The evaluator successfully connected to the TOE via TLS and captured the session ticket. The evaluator then modified the session ticket and attempt to resume this session with the TOE. The evaluator confirmed that the TOE rejected the modified ticket and issued a new session ticket for the following connection. |
| Findings: PASS |

# 4.3 Identification and Authentication (FIA)

## 4.3.1 FIA_X509_EXT.1/Rev X.509 Certificate Validation

### 4.3.1.1 TSS

325 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:** | Section 6.3.6 of the [ST] states the following:<br><br>“The TOE performs certificate validity checking for the TLS connection between the TOE and the administrative workstation.”,<br><br>“The TSF determines the validity of certificates by ensuring that the certificate and the certificate path are valid in accordance with RFC 5280.“,<br><br>“Finally, the TOE ensures the extendedKeyUsage field includes the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) for server certificates used in TLS, the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2), or the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) for OCSP certificates used for OCSP.”, and |

| | |
|---|---|
| | "Certificate validity checking takes place only on upload to the File Management page. Validity checking is done only for the TOE's own certificate and no others."<br><br>Section 6.3.6 of the [ST] also states: "Revocation checking occurs at regular intervals on certificates within the trust store. The checking is for both "Certificate" and "Certificate Authority" certificates. Revocation checking uses OCSP and occurs once every hour." |

326 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:** | Section 6.3.6 of the [ST] states: "… the TSF will validate certificate revocation status using an OCSP server in the Operational Environment. If the revocation status cannot be verified, the certificate is accepted." and "Finally, the TOE ensures the extendedKeyUsage field includes the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) for server certificates used in TLS, the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2), or the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) for OCSP certificates used for OCSP."<br><br>Section 6.3.6 of the [ST] also states: "Revocation checking occurs at regular intervals on certificates within the trust store. The checking is for both "Certificate" and "Certificate Authority" certificates. Revocation checking uses OCSP and occurs once every hour.<br><br>"If the certificate chain is incomplete, then it becomes invalidated. Revocation is performed only if a full valid chain is found." |

#### 4.3.1.2 Guidance Documentation

327 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:** | Section 3.9 of the [AGD] describes the Maintenance->Maintaining Certificate Files section in [USER] which discusses certificates and certificate authorities. It states the requirements for the certificates to be valid and specifies the conditions which would make a certificate be deemed invalid. It also references the Configuring the System and Ports->Configuring System Settings->System Settings->Features->Common Criteria Mode->Online Certificate Status Protocol section of [USER] which describes the TOE's checking and handling of certificate revocation via OCSP. |

#### 4.3.1.3 Tests

328 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)

   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.

| **High-Level Test Description** |
| --- |
| The evaluator successfully installed a certificate onto the TOE with a valid chain. The evaluator then broke the chain and attempted to install the certificate onto the TOE. The evaluator confirmed that the TOE rejected the certificate without a valid chain. |
| Findings: PASS |

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

| **High-Level Test Description** |
| --- |
| The evaluator confirmed that the TOE rejected attempts to install a certificate that was expired. The evaluator then attempted to install an otherwise valid certificate except that the certificates in the chain were expired. The evaluator confirmed that attempts to install the certificate with certificates in the chain expired were also rejected. |
| Findings: PASS |

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.

| **High-Level Test Description** |
| --- |
| The evaluator confirmed that the TOE correctly checks the revocation status of certificates via OCSP. When the OCSP response indicates a good certificate, it was accepted by the TOE. When the OCSP response indicates a revoked certificate, it was rejected by the TOE. |
| Findings: PASS |

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.

| **High-Level Test Description** |
| --- |
| The evaluator confirmed that the TOE correctly checks the revocation status of certificates via OCSP. When the OCSP response was signed by a certificate without the OCSPSigning bit set, it was rejected by the TOE. |
| Findings: PASS |

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.)

| **High-Level Test Description** |
| --- |
| The evaluator attempted to import a TLS certificate to the TOE with a modified byte in the certificate in the first eight bytes and verified that the TOE rejected the certificate. |
| Findings: PASS |

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.)

| **High-Level Test Description** |
| --- |
| The evaluator attempted to install a TLS certificate to the TOE with a modified byte in the certificate in the last byte and verified that the TOE rejected the certificate. |
| Findings: PASS |

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.)

| **High-Level Test Description** |
| --- |
| The evaluator attempted to install a TLS certificate to the TOE with a modified public key and verified that the TOE rejected the certificate. |
| Findings: PASS |

h. [Modified by TD0527] Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall conduct the following tests:

i. [Modified by TD0527] 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.

j. establish and application channel to resume the session.

| **Findings:** | This test is not applicable since the TOE does not process certificate messages. |
|---|---|

k. [Modified by TD0527] 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:** | This test is not applicable since the TOE does not process certificate messages. |
|---|---|

l. [Modified by TD0527] 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.

| **High-Level Test Description** |
|---|
| The evaluator attempted to upload an intermediate CA certificate that contains an explicitly parameterized key and confirmed that the certificate was rejected by the TOE. |
| Findings: PASS |

329 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.

330 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).

331 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).

| **High-Level Test Description** |
|---|
| The evaluator attempted to import the CA certificate onto the TOE without the basicConstraints extension and confirmed that the certificate was rejected by the TOE. |
| Findings: PASS |

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).

| **High-Level Test Description** |
|---|
| The evaluator attempted to import the CA certificate onto the TOE with the basicConstraints extension set to false and confirmed that the certificate was rejected by the TOE. |
| Findings: PASS |

332 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).

| | |
|---|---|
| **Findings:** | The TOE only uses certificates for the TLS server. |

### 4.3.2 FIA_X509_EXT.2 X.509 Certificate Authentication

#### 4.3.2.1 TSS

333 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:** | Section 6.3.6 of the [ST] states: "The TSF determines the validity of certificates by ensuring that the certificate and the certificate path are valid in accordance with RFC 5280. The TOE supports a minimum path length of three certificates for audit server certificates. While the TOE supports a minimum path length of 2 certificates for TOE certificates. In addition, the certificate path is terminated in a trusted CA certificate, the basicConstraints extension is present, and the CA flag is set to TRUE for all CA certificates. Finally, the TOE ensures the extendedKeyUsage field includes the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) for server certificates used in TLS, the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2), or the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) for OCSP certificates used for OCSP."<br><br>Section 6.3.6 of the [ST] also states: "In order to support HTTPS/TLS connectivity for the web UI interface, the TSF of all components provide the ability to generate a Certificate Request Message as specified by RFC 2986 so that its server certificate can be signed by a Certification Authority. The message includes public key, Common Name, Organization, Organizational Unit, and Country values. The certificate chain of the Certificate Response is validated by the TSF prior to being installed as the TOE's server certificate."<br><br>The section also describes the need for an OCSP responder in the operational environment required for certificate revocation checking. |

334 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:** | Section 6.3.6 of the [ST] states: "As part of the certificate validation checking, the TSF will validate certificate revocation status using an OCSP server in the Operational Environment. If the revocation status cannot be verified, the certificate is accepted." |

#### 4.3.2.2 Guidance Documentation

335 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:** | [AGD] section 3.4 describes how leaf and CA certificates can be imported onto the TOE for use. If connection cannot be established to check the validity of a certificate the TOE rejects the certificate and no further action is needed from the Security |

| | |
|---|---|
| | Administrator. [AGD] section 3.9 also states that in order for certificates to be installed on the TOE, they must be PEM formatted with a '.crt' file extension. |

#### 4.3.2.3 Tests

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

337 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.

| **High-Level Test Description** |
|---|
| The evaluator attempted to validate a certificate via OCSP. The evaluator ensured that the OCSP responder was offline and observed that the TOE attempted to reach the OCSP responder and ultimately failed and accepted the certificate. |
| Findings: PASS |

### 4.3.3 FIA_X509_EXT.3 Extended: X509 Certificate Requests

#### 4.3.3.1 TSS

338 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:** | The [ST] does not make the "device-specific information" selection. |

#### 4.3.3.2 Guidance Documentation

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:** | Section 3.9 of the [AGD] identifies the Maintenance->Maintaining Certificate Files section of the [USER] which discusses the 'generate csr' command used for generating a csr which then points to the [CLI] which outlines this command in detail. The details in the [CLI] are consistent with what is required to generate a CSR with the fields selected in the [ST]. |

#### 4.3.3.3 Tests

339 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.

| High-Level Test Description |
|---|
| The evaluator generated a certificate request on the TOE and inspected it to confirm that it contained all of the values specified in the [ST]. |
| Findings: PASS |

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.

| High-Level Test Description |
|---|
| The evaluator signed the certificate request from test 1 via an untrusted CA and attempted to import it onto the TOE. The evaluator confirmed that this attempt was rejected by the TOE. The evaluator then imported the necessary certificates to validate the signed request onto the TOE then attempted to import it again, this time successfully. |
| Findings: PASS |

## 4.4 Security management (FMT)

### 4.4.1 FMT_MOF.1/Functions Management of security functions behaviour

#### 4.4.1.1 TSS

340 For distributed TOEs see chapter 2.4.1.1.

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

341 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:** | Section 6.4.2 of the [ST] states: “The TOE restricts the ability to modify (enable/disable) transmission of audit records to an external audit server to Security Administrators.” |

#### 4.4.1.2 Guidance Documentation

342 For distributed TOEs see chapter 2.4.1.2.

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

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

| | |
|---|---|
| **Findings:** | [AGD] section 3.7 'Audit Logging' specifies how the Security Administrator can determine or modify the behavior of transmitting audit data to an external IT entity. This is the only claimed configurable setting on the TOE. |

#### 4.4.1.3 Tests

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

| **High-Level Test Description** |
|---|
| The evaluator attempted to modify the TOE's syslog server's port and IP address parameters as a user that does not have the Security Administrator role and confirmed that the changes were denied by the TOE. |
| Findings: PASS |

345 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.

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

| **High-Level Test Description** |
|---|
| The evaluator attempted to modify the TOE's syslog settings as a user that does have the Security Administrator role and confirmed that the changes were accepted by the TOE. |
| Findings: PASS |

347 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.

348 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.

349 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:** These tests are not applicable because these selections are not made by the [ST].

350 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.

351 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.

352 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:** These tests are not applicable because these selections are not made by the [ST].

353 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.

354 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:** | These tests are not applicable because these selections are not made by the [ST]. |

## 4.4.2 FMT_MTD.1/CryptoKeys Management of TSF Data

### 4.4.2.1 TSS

355 For distributed TOEs see chapter 2.4.1.1.

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

356 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:** | Section 6.4.5 of the [ST] states: "The TOE restricts generation, importation, or deletion of all cryptographic keys. To Security Administrators." |

### 4.4.2.2 Guidance Documentation

357 For distributed TOEs see chapter 2.4.1.2.

| | |
|---|---|
| **Findings:** | The TOE is not a distributed TOE. |

358 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:** | Section 3.10 of [AGD] references the Maintenance->Maintaining SSH Public Key Files and Maintenance->Maintaining Certificate Files sections in [USER] necessary to manage SSH and TLS keys and certificates. |

#### 4.4.2.3 Tests

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

| **High-Level Test Description** |
| --- |
| The evaluator attempted to install, generate and remove certificates on the TOE as a user that does not have the Security Administrator role and confirmed that the changes were rejected by the TOE. The TOE does allow users to import a certificate to the TOE; however, no functions can be executed by the user of imported certificates (i.e. install). |
| Findings: PASS |

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

| **High-Level Test Description** |
| --- |
| This is done as part of FIA_X509_EXT.3. |
| Findings: PASS |

# 5 Evaluation Activities for Security Assurance Requirements

## 5.1 ASE: Security Target

361 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:** | See above sections. |
|---|---|

362 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.<br><br>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:** | See above sections. |
|---|---|

## 5.2 ADV: Development

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

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

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

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

367 5.2.1.1 Evaluation Activity: The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

| | |
|---|---|
| **Findings:** | From section 7.2.1 of the NDcPP :<br><br>"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."<br><br>The [ST] and the AGD comprise the functional specification. If the test in [SD] cannot be completed because the [ST] or the AGD is incomplete, then the functional specification is not complete and observations are required.<br><br>During the evaluator's use of the product and its interfaces (the Web GUI, SSH CLI, local serial port), there were no areas that were deficient. |

368 5.2.1.2 Evaluation Activity: 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:** | See comments in the previous work unit. |

369 5.2.1.3 Evaluation Activity: The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

| | |
|---|---|
| **Findings:** | See comments in the previous work unit. |

## 5.3 AGD: Guidance

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

371 5.3.1.1 Evaluation Activity: 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:** | The operational guidance documentation is distributed with the TOE. The documentation is additionally available for download from the NETSCOUT Support web site. |

372 5.3.1.2 Evaluation Activity: The evaluator shall ensure that the Operational guidance is provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

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

373 5.3.1.3 Evaluation Activity: The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

| | |
|---|---|
| **Findings:** | There is only one cryptographic engine associated with the TOE. No such warning is required since there is no possibility of being able to configure a second engine. |

374 5.3.1.4 Evaluation Activity: The evaluator shall ensure the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

| | |
|---|---|
| **Findings:** | The [AGD] document covers configuration of the in-scope functionality where additional configuration might be required. In addition, section 1.3 of the [AGD] outlines the in-scope security functionality as well as the interfaces over which these functions are available. |

5.3.1.5 Evaluation Activity

375 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.

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

5) Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

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

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

| | |
|---|---|
| **Findings:** | See the findings for work unit [PP] 5.3.1.3 for configuration of the cryptographic engine.<br><br>The TOE claims published hashes instead of digital signatures. The process for obtaining the update and verifying the published hash is described in [AGD] section 2.2.<br><br>The process for manually upgrading the TOE is provided in [AGD] section 2.4 which includes an indicator as to whether the process was successful or not.<br><br>See work unit [PP] 5.3.1.4 for details as to what was covered by the EAs. |

376 5.3.2.1 Evaluation Activity: 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).

| | |
|---|---|
| **Findings:** | Please refer to work unit AGD_OPE.1-6. |

377 5.3.2.2 Evaluation Activity: The evaluator shall examine the Preparative procedures to ensure they are provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

| | |
|---|---|
| **Findings:** | There is only one operational environment claimed in the [ST].<br><br>All TOE platforms claimed in [ST] are covered by the operational guidance. This is evidenced by the single installation guide [INSTALL] and the platform equivalency. |

378 5.3.2.3 Evaluation Activity: The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.

| | |
|---|---|
| **Findings:** | See previous work unit. |

379 5.3.2.4 Evaluation Activity: The evaluator shall examine the preparative procedures to ensure they include instructions to manage the security of the TSF as a product and as a component of the larger operational environment.

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

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

The preparative procedures must:

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

b) identify TOE passwords that have default values associated with them and instructions shall be provided for how these can be changed.

| | |
|---|---|
| **Findings:** | The entire [AGD] document is designed to ensure the administrator is aware of how to configure the TOE to provide a protected administrative capability.<br><br>The [AGD] discusses the TOE's default TOE username and password. When the TOE is configured for the first time, the user is prompted to change the admin password. |

# 6 Vulnerability Assessment

381 5.6.1.1 Evaluation Activity: 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.

382 [Modified by TD0547] The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components should identify at a minimum the processors used by the TOE. Software components include applications, the operating system and other major components that are independently identifiable and reusable (outside of the TOE), for example a web server, protocol or cryptographic libraries, (independently identifiable and reusable components are not limited to the list provided in the example). This additional documentation is merely a list of the name and version number of the components and will be used by the evaluators in formulating vulnerability hypotheses during their analysis.

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

383 5.6.1.2 Evaluation Activity: The evaluator formulates hypotheses in accordance with process defined in Appendix A. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3.

| | |
|---|---|
| **Findings:** | The following sources of public vulnerabilities were considered in formulating the specific list of flaws to be investigated by the evaluators, as well as to reference in directing the evaluators to perform key-word searches during the evaluation of the TOE. Hypothesis sources for public vulnerabilities were:<br><br>-Netscout Security Advisories: https://www.netscout.com/securityadvisories<br><br>-NIST National Vulnerabilities Database (can be used to access CVE and US-CERT databases identified below): https://web.nvd.nist.gov/view/vuln/search<br><br>-Common Vulnerabilities and Exposures: http://cve.mitre.org/cve/ https://www.cvedetails.com/vulnerability-search.php<br><br>-US-CERT: http://www.kb.cert.org/vuls/html/search<br><br>-Community (Symantec) security community: https://www.securityfocus.com/<br><br>-Tenable Network Security: https://www.tenable.com/cve<br><br>-Tipping Point Zero Day Initiative: http://www.zerodayinitiative.com/advisories<br><br>-Offensive Security Exploit Database: https://www.exploit-db.com/<br><br>-Rapid7 Vulnerability Database: https://www.rapid7.com/db/vulnerabilities |

Type 1 Hypothesis searches were conducted on March 21, 2022 and included the following search terms:

Netscout nGenius

Netscout Packet Flow Switch

Packet Flow Operating Software

PFOS

Linux kernel 4.14.151

OpenSSH 8.9p1

Nginx 1.20.2

OpenSSL 1.0.2zd

Intel Atom C2538

Intel Xeon D-1518

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

There are no type-2 hypotheses identified for the NDcPP.

The evaluation team developed Type 3 flaw hypotheses in accordance with Sections A.1.3, A.1.4, and A.2, and no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.

The evaluation team developed Type 4 flaw hypotheses in accordance with Sections A.1.3, A.1.4, and A.2, and no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.