Assurance Activities Report

for

Palo Alto Networks WF-500-B Appliance running WildFire 11.1

Version 1.0

October 22, 2025

Prepared by:

Leidos Inc.
Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, MD 21046

Prepared for:

Palo Alto Networks, Inc.
3000 Tannery Way
Santa Clara, CA 95054

The Developer of the TOE:

Palo Alto Networks, Inc.
3000 Tannery Way
Santa Clara, CA 95054

The TOE Evaluation was Sponsored by:

Palo Alto Networks, Inc.
3000 Tannery Way
Santa Clara, CA 95054

Evaluation Personnel:

Justin Fisher
Anthony Apted
Greg Beaver
Josh Marciante
Allen Sant

Table of Contents

1 Introduction

This document presents results from performing evaluation activities associated with the evaluation of Palo Alto Networks WF-500-B Appliance running WildFire 11.1. This report contains sections documenting the performance of evaluation activities associated with each of the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) as specified in the Evaluation Activities for Network Device cPP, Version 3.0e, 06 December 2023 [ND-SD] and the SFRs in the Functional Package for Secure Shell (SSH), Version 1.0, 2021-05-13 [SSHPKG].

Note that, in accordance with NIAP Policy Letter #5, all cryptography in the TOE for which NIST provides validation testing of FIPS-approved and NIST-recommended cryptographic algorithms and their individual components must be NIST validated. The CCTL will verify that the claimed NIST validation complies with the NIAP-approved PP requirements the TOE claims to satisfy. The CCTL verification of the NIST validation will constitute performance of the associated assurance activity. As such, Test activities associated with functional requirements within the scope of Policy Letter #5 are performed by verification of the relevant ACVP certification and not through performance of any testing as specified in the PP or its supporting document.

1.1 Applicable Technical Decisions

The NIAP Technical Decisions referenced below apply to [CPP_ND_V3.0E]. Rationale is included for those Technical Decisions that do not apply to this evaluation.

- **TD0923**: NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN
    - This TD is applicable to the evaluation. This TD modifies a function that is supported / claimed by the TOE
- **TD0921:** NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment
    - This TD is applicable to the evaluation. This TD modifies a supported function, which has been applied to this ST [NDcPP].
- **TD0900:** Clarification to Local Administrator Access in FIA_UIA_EXT.1.3
    - This TD is applicable to the evaluation. This TD modifies a supported function, which has been applied to this ST [NDcPP].
- **TD0899:** NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2
    - This TD is applicable to the evaluation. This TD modifies test activities for TLS Clients and Server, which is supported by the TOE [NDcPP].
- **TD0886:** Clarification to FAU_STG_EXT.1 Test 6
    - This TD is applicable to the evaluation. This TD adds an application note to clarify a test activity that applies to the TOE [NDcPP].

- **TD0880:** Removal of Duplicate Selection in FMT_SMF.1.1
  - This TD is applicable to the TOE. This TD removes a duplicate selection [NDcPP].
- **TD0879:** Correction of Chapter Headings in CPP_ND_V3.0E
  - The TD is applicable to the evaluation. This TD modifies the chapter numbering and has been applied [NDcPP].
- **TD0868**: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8
  - This TD is not applicable to the TOE. The TOE does not implement IPsec.
- **TD0836:** FPT_TST_EXT.1.1 in CPP_ND_V3.0E has been modified
  - This TD is applicable to the TOE. The TD modifies the FPT_TST_EXT.1.1 SFR and the AAR evaluation activities have been modified.

The NIAP Technical Decisions referenced below apply to [SSHPKG]. Rationale is included for those Technical Decisions that do not apply to this evaluation.

- **TD0682:** Addressing Ambiguity in FCS_SSHS_EXT.1 Tests
  - This TD is applicable to the TOE. This TD updates test evaluation activities that apply to the TOE.
- **TD0695:** Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package
  - This TD is applicable to the TOE. The TD updates FCS_COP.1, which includes AES-CTR.
- **TD0732:** FCS_SSHS_EXT.1.3 Test 2 Update
  - This TD is applicable to the TOE. The TD updates test evaluation activities that apply to the TOE.
- **TD0777:** Clarification to Selections for Auditable Events for FCS_SSH_EXT.1
  - This TD is applicable to the TOE because it logs failure to establish SSH connection.
- **TD0909:** Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0
  - This TD is applicable to the TOE as it supports public keys.

1.2 Evidence

[ST] Palo Alto Networks WF-500-B Appliance running WildFire 11.1 Security Target, Version 1.0, October 22, 2025

[CCECG] Palo Alto Networks Common Criteria Evaluated Configuration Guide (CCECG) for WildFire 11.1, Revision Date: October 22, 2025

[Admin] WildFire Appliance Administration, Version 11.1, 23 August 2023

[WF-500-B HW REF] WF-500-B Appliance Hardware Reference, March 21, 2023

[Test] Palo Alto Networks WF-500-B WildFire 11.1 Common Criteria Test Report and Procedures For Network Device collaborative PP Version 3.0e, Version 1.0, October 22, 2025

[VA] Palo Alto Networks WF-500-B WildFire 11.1 Vulnerability Assessment, Version 1.0, October 22, 2025

1.3 Conformance Claims

Common Criteria Versions

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction, Version 3.1, Revision 5, dated April 2017.
- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Revision 5, dated April 2017.
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Revision 5, dated April 2017.

Common Evaluation Methodology Versions

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 5, dated April 2017.

Protection Profiles

- PP Reference: collaborative Protection Profile for Network Devices, Version 3.0e, 6-December-2023 [NDcPP]
- Package Reference: Functional Package for Secure Shell (SSH) Version 1.0 13-May-2021 [SSHPKG]

1.4 SAR Evaluation

The following Security Assurance Requirements (SARs) were evaluated during the evaluation of the TOE:

SAR	Verdict
ASE_CCL.1	Pass
ASE_ECD.1	Pass
ASE_INT.1	Pass
ASE_OBJ.1	Pass
ASE_REQ.1	Pass
ASE_SPD.1	Pass
ASE_TSS.1	Pass
ADV_FSP.1	Pass

AGD_OPE.1	Pass
AGD_PRE.1	Pass
ALC_CMC.1	Pass
ALC_CMS.1	Pass
ALC_FLR.3	Pass
ATE_IND.1	Pass
AVA_VAN.1	Pass

The evaluation work units are listed in the proprietary ETR. The evaluators note per the PP evaluation activities that many of the SARs were successfully evaluated through completion of the associated evaluation activities present in the claimed PP.

2 Security Functional Requirement Evaluation Activities

This section describes the evaluation activities associated with the SFRs defined in the ST and the results of those activities as performed by the evaluation team. The evaluation activities are derived from [ND-SD] and [SSHPKG] and modified by applicable NIAP Technical Decisions. Evaluation activities for SFRs not claimed by the TOE have been omitted.

Evaluator notes, such as changes made as a result of NIAP Technical Decisions, are highlighted in bold text, as are changes made as a result of NIAP Technical Decisions. Bold text is also used within evaluation activities to identify when they are mapped to individual SFR elements rather than the component level.

2.1 Security Audit (FAU)

2.1.1 FAU_GEN.1 Audit Data Generation

2.1.1.1 TSS Activities

> 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.

Section 6.1 of [ST] asserts under FAU_GEN.1 that the key or certificate name (if the key is embedded in a certificate or certificate request) is logged for any auditable event that relates to key operations.

> 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.

The TOE is not distributed so this evaluation activity is not applicable.

2.1.1.2 Guidance Activities

> 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).

Section 4 of [CCECG] includes a table listing of the auditable events for NDcPP.

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.

[CCECG] section 4 identifies sample audit records for administrative configuration of the TOE, as determined by the management functions specified in FMT_SMF.1.

2.1.1.3 Test Activities

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different identity and authentication (I&A) mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are cPP_ND_v3.0e-SD, 06-Dec-2023 16 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.

The evaluator performed the auditable actions, either independently or as part of an evaluation activity, to generate all audit records on the TOE and confirmed that they were generated in the format specified in guidance for each auditable action.

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.

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

This evaluation activity is N/A for this evaluation. The TOE is not distributed.

2.1.2 FAU_GEN.2 User Identity Association

2.1.2.1 TSS Activities

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

2.1.2.2 Guidance Activities

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

2.1.2.3 Test Activities

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

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.

This evaluation activity is N/A for this evaluation. The TOE is not distributed.

2.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

2.1.3.1 TSS Activities

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.

Section 6.1 of [ST] states that the TSF sends audit data to a remote syslog server over TLS.

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.

Section 6.1 of [ST] states that the TOE is a standalone TOE that stores audit data locally. The TSS also states when a log reaches the maximum size (45 MB is the minimum that can be set), the TOE drops new audit data. It also states that audit records are protected against unauthorized access through the fact that the TSF includes no interface to interact directly with the records on the file system.

[ST] does not identify a specific size for audit storage because the customer can customize the number and storage capacity of hard disks.

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

[ST] section 6.1 indicates that the TOE can be configured to send generated audit records to an external Syslog server in real-time using TLS. When configured to send audit records to a syslog server, audit records are also written to the external syslog as they are written locally to the internal logs.

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).

This evaluation activity is not applicable; the TOE is not distributed.

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

[ST] selects “Drop new audit data”, Section 6.1 of [ST] states that when the log file reaches its maximum size, the TOE drops new audit data.

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

Section 6.1 of [ST] states that audit data is stored across multiple logs files present on the file system of the TOE. As the data is stored in files present on the system it has been concluded that the logs would be stored in a persistent manner.

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

Section 6.1 of [ST] states that the TOE stores the audit records locally and protects them from unauthorized deletion by allowing only users in the pre-defined Audit Administrator role to access the audit trail with delete privileges.

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

[ST] selects “drop new audit data” so no further description in the TSS is necessary.

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.

This evaluation activity is N/A for this evaluation. The TOE is not distributed.

2.1.3.2 Guidance Activities

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.

Section 3 of [CCECG] lists the required ports that must be available to support the various communications interfaces. For the audit server (syslog over TLS), it indicates that TCP port 6514 must be open.

Section 6.8.1 of [CCECG] states the TOE can be configured to forward generated audit records to an external syslog server in real-time and provides guidance to configure the TOE to establish a trusted channel to the external syslog server in order to forward the audit records over the trusted channel. Guidance is provided for establishing a trusted channel using TLS v1.2. Section 6.8.1 of [CCECG] also describes requirements for configuring a Syslog Server Profile, importing certificates, and configuring the External Syslog-ng Server.

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

Section 6.8.1 of [CCECG] states the TOE can be configured to forward generated audit records to an external syslog server in real-time. Audit records are converted and forwarded to the external syslog as they are locally written to the log files.

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

Section 4 of [CCECG] states the TOE has an internal log database that can be used to store and review audit records locally. Once the audit log is full, the newest audit records are dropped.

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

Storage size is not configurable.

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

The ST only makes on selection for FAU_STG_EXT.1.5, therefore this is not applicable.

2.1.3.3 Test Activities

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

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

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.

The evaluator configured the TOE to send audit records to an audit server protected by TLS. The evaluator confirmed that the audit server received the audit records and the packets were protected via TLS in transit.

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

This test is not applicable as the TOE is not a distributed TOE.

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

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

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

The evaluator observed the oldest audit records present on the TOE, then rebooted the device and verified the oldest audit records were still present as the TOE implements a persistent log store.

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

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

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

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

The evaluator performed actions to generate audit events until the local audit trail was full and confirmed that new audit data was dropped, consistent with the ST claim.

Test 5: For distributed TOEs, for the local storage according to FAU_STG_EXT.1.4, Test 1 specified above shall be applied to all TOE components that store audit data locally. For all TOE components that store audit data locally and comply with FAU_STG_EXT.2, Test 2 specified above shall be applied. The evaluator shall verify that the transfer of audit data to an external audit server is implemented.

This test is not applicable as the TOE is not a distributed TOE.

TD0886

An Application Note is appended to Test 6 for FAU_STG_EXT.1 in CPP_ND_V3.0E-SD as follows:

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

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

The evaluator verified the provides the ability to view the audit records locally.

2.1.4 FAU_STG.1 Protected Audit Trail Storage

2.1.4.1 TSS Activities

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

Section 6.1 of [ST] states that the TOE does not provide an interface where a user can modify the audit records, thus it prevents modification to the audit records. When a log reaches the maximum size (at least 45 MB), new audit is dropped. Maximum disk space is dependent on the customer's installation as it depends on the number of hard drives installed on the system.

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

This evaluation activity is N/A for this evaluation. The TOE is not distributed.

2.1.4.2 Guidance Activities

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

There is no TSF configuration that is required to protect locally-stored audit data; the "Before Installation You Must" section of the guidance documentation provides instructions to ensure the TOE's physical and logical environment is restricted in such a manner to minimize unauthorized access to the TSF in general, which includes stored audit data.

2.1.4.3 Test Activities

Test 1: The evaluator shall attempt to access the audit trail without authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all) and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to access the audit trail can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

The evaluator verified the TOE does not grant unauthenticated users the ability to modify or delete audit records.

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

The evaluator verified the TOE provides the ability for an authenticated Security Administrator to delete/clear the audit record store.

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

This portion of the test activity is not applicable as the TOE is not a distributed TOE.

2.2 Cryptographic Support (FCS)

Section 6.2 of [ST] identifies the TOE's ACVP certificates with respect to its cryptographic claims that require them. This has been reproduced below, with the relevant SFRs added as per Labgram #108.

As per Labgram #108, the full justification and evidence for why these certificates are sufficient to meet the evaluation activity requirements for the SFRs in question has been included in the Certificate Reporting section in the ETR.

Functions	Standards	Certificates
Asymmetric Key Generation (FCS_CKM.1)		
ECC key pair generation (NIST curves P-256, P-384, P-521) *Note that TLS and SSH each use any of P-256, P-384, or P-521, and certificate generation uses only P-256 or P-384.*	FIPS PUB 186-4	Safe Primes Key Generation #A3453 ECDSA #A3453 RSA #A3453
RSA key generation (key sizes 2048, 3072, 4096 bits) *Note that TLS uses 2048-bit RSA keys while certificate and SSH RSA key pair generation use 2048, 3072, or 4096-bit keys.*	FIPS PUB 186-4	
FFC Schemes using Diffie-Hellman groups that meet the following: RFC 3526 and SP 800-56Ar3 (2048-bit MODP)	RFC 3526 NIST SP 800-56Ar3	

Functions	Standards	Certificates
Cryptographic Key Establishment (FCS_CKM.2)		
ECDSA based key establishment (NIST P-256, P-384, P-521)	NIST SP 800-56Ar3	ECC #A3453 FFC #A3453
FFC-based Key establishment scheme using Diffie-Hellman groups that meet the following: RFC 3526 and SP 800-56Ar3 (2048-bit MODP)	RFC 3526 NIST SP 800-56Ar3	
AES Data Encryption/Decryption (FCS_COP.1/DataEncryption)		
AES CBC, CTR, GCM (128, 256 bits)	AES as specified in ISO 18033-3 CBC as specified in ISO 10116 CTR as specified in ISO 10116 GCM as specified in ISO 19772	AES #A3453
Signature Generation and Verification (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (rDSA) (modulus 2048, 3072, 4096)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	RSA #A3453
ECDSA (NIST curves P-256, P-384, and P-521)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" P-256, P-384, ISO/IEC 14888-3, Section 6.4	ECDSA #A3453
Cryptographic Hashing (FCS_COP.1/Hash)		
SHA-1, SHA-256, SHA-384 and SHA-512 (digest sizes 160, 256, 384 and 512 bits)	ISO/IEC 10118-3:2004	SHS #A3453
Keyed-hash Message Authentication (FCS_COP.1/KeyedHash)		
• HMAC-SHA-1 (key size 160 bits and digest size 160 bits) • HMAC-SHA-256 (key size 256 bits and digest size 256 bits) • HMAC-SHA-384 key size 384 bits and digest size 384 bits) • HMAC-SHA-512 (key size 512 bits and digest size 512 bits)	ISO/IEC 9797-2:2011	HMAC #A3453

Functions	Standards	Certificates
Random Bit Generation (FCS_RBG_EXT.1)		
CTR_DRBG (AES) from a platform-based noise source of 256 bits of non-determinism	ISO/IEC 18031:2011	DRBG #A3453

2.2.1 FCS_CKM.1 Cryptographic Key Generation

2.2.1.1 TSS Activities

> 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.

Section 6.2 of [ST] lists the key sizes and schemes supported by the TOE:

- FFC key pair generation (key size 2048 bits)
- ECC key pair generation (NIST curves P-256, P-384, P-521)
    - Note that TLS and SSH each use any of P-256, P-384, or P-521, and certificate generation uses only P-256 or P-384.
- RSA key generation (key sizes 2048, 3072, 4096 bits)
    - Note that TLS uses 2048-bit RSA keys while certificate and SSH RSA key pair generation use 2048, 3072, or 4096-bit keys.

The TOE can be configured as a TLS server for mutual certificate-based authentication for secure connections. The key agreement parameters of the server key exchange message consist of the key establishment parameters generated by the TOE: Diffie-Hellman parameters with a key size 2048 bits (group 14), ECDSA implementing NIST curves secp256r1 and secp384r1.

2.2.1.2 Guidance Activities

> 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.

Section 6.2 of [CCECG] describes how to enable FIPS-CC Mode on the TOE and states that it is required for the evaluated configuration. This process will configure cryptographic parameters to ensure that only the required key generation schemes are used for trusted channel communications.

For certificates, section 6.8.1 of [CCECG] describes how to generate a CSR under the heading “CSR Generation.” The command outlined in this section identifies the supported algorithms and key sizes for the certificate consistent with the claims made in [ST] about the key generation algorithms used for X.509 certificates.

2.2.1.3 Test Activities

Modified by TD0921

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

Performed in accordance with NIAP Policy Letter #5.

Key Generation for Elliptic Curve Cryptography (ECC)

Performed in accordance with NIAP Policy Letter #5.

Key Generation for Finite-Field Cryptography (FFC)

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the ACVP certifications verifying asymmetric key generation, as follows.

Functions	Standards	Certificates
Asymmetric Key Generation (FCS_CKM.1)		
ECC key pair generation (NIST curves P-256, P-384, P-521) *Note that TLS and SSH each use any of P-256, P-384, or P-521, and certificate generation uses only P-256 or P-384.*	FIPS PUB 186-4	Safe Primes Key Generation #A3453 ECDSA #A3453 RSA #A3453
RSA key generation (key sizes 2048, 3072, 4096 bits) *Note that TLS uses 2048-bit RSA keys while certificate and SSH RSA key pair generation use 2048, 3072, or 4096-bit keys.*	FIPS PUB 186-4	
FFC Schemes using Diffie-Hellman groups that meet the following: RFC 3526 and SP 800-56Ar3 (2048-bit MODP)	RFC 3526 NIST SP 800-56Ar3	

FFC Schemes using "safe-prime" groups

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

2.2.2 FCS_CKM.2 Cryptographic Key Establishment

2.2.2.1 TSS Activities

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.

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

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

Sections 6.2 and 6.7 of [ST] identifies that the TOE uses the following key establishment schemes for each cryptographic service mapped to a claimed protocol:

- SSH:
  - FIPS 186-4 (ECC) – 256, 384, 521 bits
    - Provides a communication path between itself and authorized remote administrators.
- TLS:
  - FIPS 186-4 FFC (safe primes) –2048-bit MODP
  - FIPS 186-4 ECDHE (ECC) – 256, 384, 521 bits.
    - Sends audit records to an external syslog server using TLS in real-time. The TOE permits the TSF to initiate communication with the syslog server using the TLS trusted channel. The TOE (TLS server) can also communicate with the Palo Alto Networks firewalls via TLS.

The supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1.

2.2.2.2 Guidance Activities

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

Section 6.2 of [CCECG] describes how to enable FIPS-CC Mode on the TOE and states it is required for the evaluated configuration. This process configures cryptographic parameters to ensure the TOE uses only the key generation schemes specified in [ST] for trusted channel communications.

2.2.2.3 Test Activities

Key Establishment Schemes

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

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

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the ACVP certifications verifying SP 800-56A key establishment schemes, as follows.

Algorithm	Tested Capabilities	Certificates
ECDSA based key establishment (NIST P-256, P-384, P-521)	NIST SP 800-56Ar3	ACVP #A3453 KAS-ECC-SSC
FFC-based Key establishment scheme using Diffie-Hellman groups that meet the following: RFC 3526 and SP 800-56Ar3 (2048-bit MODP)	RFC 3526 NIST SP 800-56Ar3	ACVP #A3453 KAS-FFC-SSC

RSA-based Key Establishment

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.

The ST does not select RSA-based key establishment schemes in FCS_CKM.2, so this activity is not applicable.

FFC Schemes using "safe-prime" groups

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.

Refer to test activities for FTP_TRP.1/Admin.

2.2.3 FCS_CKM.4 Cryptographic Key Destruction

2.2.3.1 TSS Activities

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

Section 6.2 of [ST] lists all keys/CSPs used by the TOE by their function (e.g. RSA/ECDSA private keys, SSH session keys) and describes their usage and composition.

Section 6.2 of [ST] states that Private cryptographic keys, plaintext cryptographic keys, and all other critical security parameters stored in intermediate locations in volatile memory for the purposes of transferring the key/critical security parameters (CSPs) to another location are zeroized immediately following the transfer. Zeroization is done by overwriting the storage location with a random pattern, followed by a read-verify. Note that plaintext cryptographic session keys (e.g., TLS encryption keys, SSH session keys) and CSPs (e.g., TLS Pre-Master secret, ECDHE/DHE private components) are only ever stored in volatile memory.

For non-volatile memories other than EEPROM and Flash, the zeroization is executed by overwriting three times using a different alternating data pattern of ones and zeros each time. This includes the SSD storage. This includes all CSPs that are not stored in volatile memory such as private keys, KEK, hashed passwords, and entropy seeds. Only the KEK is stored in plaintext. It is used to encrypt all the private keys and other sensitive data. When a new KEK is generated, the old KEK is destroyed via key store APIs that overwrites the old KEK. The KEK is erased when the administrator initiates the zeroization function, which overwrites the KEK three or more times using an alternating pattern of ones and zeroes. Destruction of all encrypted stored keys is accomplished indirectly through destruction of the KEK that encrypted them.

For volatile memory and non-volatile EEPROM and Flash memories, the zeroization is executed by a single direct overwrite consisting of a pseudo random pattern, followed by a read-verify. Sensitive data in volatile memory includes session keys such as encryption keys, integrity keys, and the Pre-Master secret.

Section 6.2 of [ST] lists each type of key/CSP used by the TOE, the cryptographic algorithm the key/CSP acts as input or output data for, how it is generated/used, the type of storage medium it resides on, and its method of destruction. In all cases, the key storage locations and destruction methods are consistent with the claims made in the SFR.

> 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).

Section 6.2 of [ST] states that the TSF invokes key store APIs to erase the KEK in non-volatile storage, which has the effect of erasing all keys that are protected by the KEK.

> 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.

Section 5.2.2 of [ST] does not select 'destruction of reference' or 'invocation of an interface' so this evaluation activity is not applicable to the TOE.

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.

Section 6.2 of [ST] identifies the KEK as a 256-bit AES key that encrypts all key data stored in non-volatile memory in CBC mode. The KEK is destroyed using a three or more pass alternating overwrite after a new KEK (or "Firmware Content Encryption Key") is generated.

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.

The evaluators examined [ST] and observed that no exceptions to the behavior described by FCS_CKM.4.1 have been identified, so it can be assumed that this behavior is followed in all cases.

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.

[ST] does not claim any instances where key/CSP data is overwritten with a value that does not contain any CSP.

2.2.3.2 Guidance Activities

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.

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

[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).]

[CCECG] Section 6.2 Enable FIPS-CC Mode (Required) states that enabling FIPS-CC Mode will completely zeroize the TOE and the KEK (i.e., Master Key); all configurations and logs will be erased permanently.

The administrator must enable FIPS-CC Mode.

[CCECG] does not define any circumstances that would cause key destruction to be delayed or prevented. The evaluators reviewed the TSS and test evidence and observed that no such cases should be expected.

2.2.3.3 Test Activities

None

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

2.2.4.1 TSS Activities

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.

[ST] section 6.2 identifies that the TOE supports AES with key sizes 128 bits and 256 bits across CBC, CTR, and GCM modes.

2.2.4.2 Guidance Activities

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.

Section 6.2 of [CCECG] describes how to enable FIPS-CC Mode on the TOE and states that it is required for the evaluated configuration. This process will restrict the TLS version and cipher suites to the approved ones claimed in the ST, including the allowed data encryption modes and key sizes.

Section 6.4 of [CCECG] instructs the administrator how to configure the data encryption algorithms, including modes and key sizes, to be used by the TOE in SSH connections.

2.2.4.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the ACVP certifications verifying AES encryption and decryption, as follows.

Algorithm	Tested Capabilities	Certificates
AES as specified in ISO 18033-3, CBC as specified in ISO 10116	Direction: Decrypt, Encrypt Key Length: 128, 256	ACVP #A3453 AES-CBC
AES as specified in ISO 18033-3, GCM as specified in ISO 19772	Direction: Decrypt, Encrypt IV Generation: Internal IV Generation Mode: 8.2.1 Key Length: 128, 192, 256 Tag Length: 64, 96, 104, 112, 120, 128 IV Length: 96 Payload Length: 0-1024 Increment 8 AAD Length: 8-1024 Increment 8	ACVP #A3453 AES-GCM
AES as specified in ISO 18033-3, CTR as specified in ISO 10116	Direction: Decrypt, Encrypt Key Length: 128, 256 Payload Length: 8-128 Increment 8	ACVP #A3453 AES-CTR

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

2.2.5.1 TSS Activities

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

[ST] section 6.2 states that RSA (2048/3072/4096 bit modulus) and ECDSA (P-256/P-384/P-521 curves) are supported for digital signatures.

2.2.5.2 Guidance Activities

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.

Section 6.2 of [CCECG] describes how to enable FIPS-CC Mode on the TOE and states that it is required for the evaluated configuration. This process will restrict the TLS version and cipher suites to the approved ones claimed in the ST, including the allowed cryptographic algorithms and key sizes used by the TOE for signature services.

Section 6.7 of [CCECG] instructs the administrator how to configure SSH public key authentication, including the public key algorithm and key sizes.

2.2.5.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the ACVP certifications verifying digital signature services, as follows.

Algorithm	Tested Capabilities	Certificates
For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	**RSA Signature Generation** Signature Type: PKCS 1.5 Modulo: 2048 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 3072 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 4096 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512	ACVP #A3453 RSA SigGen (FIPS186-4)
	Signature Type: PKCSPSS Modulo: 2048 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64 Modulo: 3072 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64 Modulo: 4096 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64	

Algorithm	Tested Capabilities	Certificates
	RSA Signature Verification Signature Type: PKCS 1.5 Modulo: 2048 Hash Algorithm: SHA1 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 3072 Hash Algorithm: SHA1 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 4096 Hash Algorithm: SHA1 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Signature Type: PKCPSS Modulo: 2048 Hash: SHA1; Salt Length: 20 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64 Modulo: 3072 Hash: SHA1; Salt Length: 20 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64 Modulo: 4096 Hash: SHA1; Salt Length: 20 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64	ACVP #A3453 RSA SigVer (FIPS186-4)

Algorithm	Tested Capabilities	Certificates
ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4	**ECDSA Signature Generation** Curve: P-256, P-384, P-521 Hash Algorithm: SHA2-256, SHA2-384, SHA2-512 **ECDSA Signature Verification** Curve: P-256, P-384, P-521 Hash Algorithm: SHA-1, SHA2-256, SHA2-384, SHA2-512	ACVP #A3453 ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)

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

2.2.6.1 TSS Activities

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.

Section 6.2 of [ST] indicates that the hash function is associated with digital signature generation/verification and with HMAC.

Section 6.2 of [ST] discusses the use of digital signature generation/verification and HMAC in conjunction with trusted communications and the use of SHA-256 with protection of user passwords.

Section 6.5 of [ST] identifies the TOE's use of digital signature verification for trusted updates.

2.2.6.2 Guidance Activities

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

Section 6.2 of [CCECG] describes how to enable FIPS-CC Mode on the TOE and states that it is required for the evaluated configuration. This process will configure cryptographic parameters to ensure that only the required hash algorithms are used for trusted channel communications.

If creating a certificate, section 6.8.1 of [CCECG] provides the instructions to generate a certificate that uses a supported hash algorithm.

2.2.6.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the ACVP certifications verifying cryptographic hashing, as follows.

Algorithm	Standard	Certificates

SHA-1, SHA-256, SHA-384 and SHA-512 (digest sizes 160, 256, 384 and 512 bits)	ISO/IEC 10118-3:2004	ACVP #A3453 SHA-1, SHA2-256, SHA2-384, SHA2-512

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

2.2.7.1 TSS Activities

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.

Table 6 of [ST] includes a list of the HMAC functions that lists the key size and digest size (which implicitly identifies the hash function, block size, and output MAC length).

2.2.7.2 Guidance Activities

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.

Section 6.2 of [CCECG] describes how to enable FIPS-CC Mode on the TOE and states that it is required for the evaluated configuration. This process will configure cryptographic parameters to ensure that only the required keyed-hash algorithms are used for trusted channel communications.

Section 6.4 of [CCECG] additionally states the command the administrator can use to restrict HMAC algorithms utilized by the TOE for the SSH protocol through the command ‘set deviceconfig system ssh profiles mgmt.-profiles server-profiles <profile_name> mac <hmac-name>’ CLI command.

2.2.7.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the ACVP certifications verifying cryptographic keyed hashing, as follows.

Functions	Standards	Certificates
• HMAC-SHA-1 (key size 160 bits and digest size 160 bits) • HMAC-SHA-256 (key size 256 bits and digest size 256 bits) • HMAC-SHA-384 (key size 384 bits and digest size 384 bits)	ISO/IEC 9797-2:2011	ACVP #A3453 HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512

Functions	Standards	Certificates
• HMAC-SHA-512 (key size 512 bits and digest size 512 bits)		

2.2.8 FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)

2.2.8.1 Evaluation Activity

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

The vendor produced a proprietary Entropy Analysis Report (EAR) that the evaluators determined was suitable to meet the requirements specified in Appendix D of [NDcPP].

2.2.8.2 TSS Activities

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.

Section 6.2 of [ST] states that the TSF uses an AES-256 CTR_DRBG that receives entropy from a hardware source (identified in the proprietary EAR), and states that the min-entropy of the combined seed value is 256 bits.

2.2.8.3 Guidance Activities

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

Section 6.2 of [CCECG] states that enabling FIPS-CC mode configures the DRBG to use the algorithm claimed in [ST].

2.2.8.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the ACVP certification verifying deterministic random bit generation, as follows.

Algorithm	Tested Capabilities	Certificates
CTR_DRBG in accordance with ISO/IEC 18031:2011	Counter DRBG Mode: AES-256	ACVP #A3453 Counter DRBG

2.2.9 FCS_SSH_EXT.1 SSH Protocol

2.2.9.1 TSS Evaluation Activity

FCS_SSH_EXT.1.1
The evaluator shall ensure that the selections indicated in the ST are consistent with selections in this and subsequent components. Otherwise, this SFR is evaluated by activities for other SFRs.

The evaluator examined the SFR and verified that the SFR selections of the RFCs were correct. The selections were consistent with the FCS_SSH_EXT.1 elements and the TSS.

FCS_SSH_EXT.1.2
The evaluator shall check to ensure that the authentication methods listed in the TSS are identical to those listed in this SFR component; and, ensure if password-based authentication methods have been selected in the ST then these are also described; and, ensure that if keyboard-interactive is selected, it describes the multifactor authentication mechanisms provided by the TOE.

[ST] Section 6.2 identifies the authentication public-key algorithms ssh-rsa, rsa-sha2-256, and rsa-sha2-512 are permitted in the evaluated configuration. In addition, to the public-key (RSA) authentication, password-based authentication can be configured with password-based being the default method. For password, the TOE verifies the user identity when the username is entered.

The TSS descriptions agree with the SFR.

FCS_SSH_EXT.1.3
The evaluator shall check that the TSS describes how "large packets" are detected and handled.

[ST] Section 6.2 states that SSH packets are limited to 262,105 bytes and any packet over that size will be dropped (i.e., not processed farther and buffer containing the packet will be freed).

FCS_SSH_EXT.1.4
The evaluator will check the description of the implementation of SSH in the TSS to ensure the encryption algorithms supported are specified. The evaluator will check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

[ST] Section 6.2 states that the TOE supports the AES encryption/decryption algorithm (in CBC, CTR, or GCM mode) with key sizes of 128 or 256 bits. The encryption algorithms specified in the TSS are identical to those listed in the SFR.

FCS_SSH_EXT.1.5
The evaluator will check the description of the implementation of SSH in the TSS to ensure the hashing algorithms supported are specified. The evaluator will check the TSS to ensure that the hashing algorithms specified are identical to those listed for this component.

[ST] Section 6.2 states that the TOE supports HMAC-SHA-256, HMAC-SHA-512, implicit MAC (aes128-gcm@openssh.com and aes256-gcm@openssh.com) hashing algorithms for integrity. The hashing algorithms identified in the TSS are identical to those identified in the SFR component.

FCS_SSH_EXT.1.6

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

[ST] Section 6.2 states that the TOE implements the key exchange algorithms ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 to establish a shared secret with the peer. The shared secret establishment algorithms specified in the TSS are identical to those listed in the SFR.

FCS_SSH_EXT.1.7

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

[ST] Section 6.2 states that the TOE implements SSH KDF as defined in RFC 5656 section 4 for use in ECC key establishment schemes.

FCS_SSH_EXT.1.8

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

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

a. An argument describing this hardware-based limitation and
b. Identification of the hardware components that form the basis of such argument.

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

[ST] Section 6.2 states that the TOE manages a tracking mechanism for each SSH session so that it can initiate a new key exchange (rekey) when either a configurable amount of data (10 – 4000 MBs) or time (10 – 3600 seconds) has passed, whichever threshold occurs first. In the evaluated configuration, the administrator should not configure the SSH data rekey threshold to be more than 1024 MBs that apply to both transmitted and received data.

2.2.9.2 Guidance Evaluation Activity

FCS_SSH_EXT.1.1

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

There are no guidance evaluation activities for this component.

FCS_SSH_EXT.1.2

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

Section 6.7 of [CCECG] provides the guidance to configure the SSH authentication mechanisms.

FCS_SSH_EXT.1.3

None listed.

There are no guidance evaluation activities for this component.

FCS_SSH_EXT.1.4

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

Section 6.4 "Configure SSH Encryption and Integrity Algorithms (Required)" of [CCECG] contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

FCS_SSH_EXT.1.5

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

Section 6.4 "Configure SSH Encryption and Integrity Algorithms (Required)" of [CCECG] contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

FCS_SSH_EXT.1.6

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

Section 6.4 "Configure SSH Encryption and Integrity Algorithms (Required)" of [CCECG] contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

FCS_SSH_EXT.1.7

None listed.

There are no guidance evaluation activities for this component.

FCS_SSH_EXT.1.8

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

Section 6.6 “Configure SSH Rekey Interval (Required) of [CCECG] contains instructions to the administrator on how configure the rekey interval. When FIPS-CC mode is enabled, the SSH rekeying will occur approximately at 1 hour of time or after 1 GB of data has been transmitted, whichever occurs first.

2.2.9.3 Test Evaluation Activity

FCS_SSH_EXT.1.1

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

FCS_SSH_EXT.1.2

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

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

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

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

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

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

For Part A: The evaluator verified the TOE’s SSH server only offered a SSH client the claimed authentication methods.

For Part B: This portion is not applicable as the TOE does not utilize X509 Certificates for SSH client authentication.

FCS_SSH_EXT.1.2

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

a. The evaluator shall initiate a SSH session using the authentication method configured and verify that the session is successfully established.
b. Next, the evaluator shall use bad authentication data (e.g. incorrectly generated certificate or incorrect password) and ensure that the connection is rejected.

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

This test is not applicable as the TOE does not claim SSH client functionality.

FCS_SSH_EXT.1.2

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

a. The evaluator shall configure the Client with an authentication method not supported by the Server.
b. The evaluator shall verify that the connection fails.

This test is not applicable as the TOE does not claim SSH client functionality.

FCS_SSH_EXT.1.2

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

This test is not applicable as the TOE does not claim SSH client functionality.

FCS_SSH_EXT.1.3

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

The evaluator verified the TOE’s SSH implementation accepts a packet that is at the defined maximum allowable packet size.

Modified by TD0732

FCS_SSH_EXT.1.3

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

a. The evaluator shall establish a successful SSH connection with the peer.
b. Next the evaluator shall craft a packet that is ~~one byte~~ **slightly** larger than the maximum size specified in this component and send it through the established SSH connection to the TOE. **The packet should not be greater than the maximum packet**

size + 16 bytes. If the packet is larger, the evaluator shall justify the need to send a larger packet.

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

The evaluator verified the TOE drops SSH packets which are larger than the maximum allowed packet size.

FCS_SSH_EXT.1.4

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

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

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

The evaluator verified the TOE only offers the claimed algorithms for a SSH connection to be negotiated with.

FCS_SSH_EXT.1.4

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

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

The evaluator verified that the TOE terminates the connection when the SSH connection is terminated.

FCS_SSH_EXT.1.4

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

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

The evaluator attempted to establish a SSH connection which uses a cipher algorithm not claimed by the TOE and verified the TOE did not accept the connection.

FCS_SSH_EXT.1.5

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

The evaluator verified that the values provided in FCS_SSH_EXT.1.4 are consistent with the claimed values in FCS_SSH_EXT.1.5.

FCS_SSH_EXT.1.5

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

The evaluator attempted to establish a SSH connection which uses a hash algorithm not claimed by the TOE and verified the TOE did not accept the connection.

FCS_SSH_EXT.1.6

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

The evaluator verified that the values provided in FCS_SSH_EXT.1.4 are consistent with the claimed values in FCS_SSH_EXT.1.6.

FCS_SSH_EXT.1.6

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

The evaluator attempted to establish a SSH connection which uses a key exchange algorithm not claimed by the TOE and verified the TOE did not accept the connection.

FCS_SSH_EXT.1.7

None listed.

FCS_SSH_EXT.1.8

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

The SSH test tool utilized by the lab does not implement SSH rekey initiation logic but will obey rekey initiations sent by the peer.

FCS_SSH_EXT.1.8

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

The evaluator established a SSH connection with the TOE and sent just enough data to keep the session alive. The evaluator verified the TOE rekeyed the session once the session lifetime was reached.

FCS_SSH_EXT.1.8

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

The evaluator established a SSH connection with the TOE and induced the TOE to send vastly more data than input to the TOE. The evaluator verified the TOE rekeyed the session once the session lifetime was reached.

FCS_SSH_EXT.1.8

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

The evaluator established a SSH connection with the TOE and sent vast amounts of data at the TOE. The evaluator verified the TOE rekeyed the session once the session lifetime was reached.

2.2.10 FCS_SSHS_EXT.1 SSH Server Protocol

2.2.10.1 TSS Activities

No activities.

2.2.10.2 Guidance Activities

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

[CCECG] sections 6.4 through 6.7 provides the instructions to configure the SSH server.

2.2.10.3 Test Activities

Modified by TD0682

The evaluator shall ~~**repeat Test 1 and Test 2 from FCS_SSH_EXT.1.4 for each of the authentication mechanisms supported by the TOE.**~~ perform the following tests:

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

The evaluator verified that the TOE offered only the claimed host key algorithms to a client for the SSH server authentication.

Modified by TD0682

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

The evaluator verified that the TOE could successfully use each of the claimed algorithms for the SSH server authentication and presented a value correct for the algorithm to the client.

Modified by TD0682

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

The evaluator verified the TOE did not complete a connection when the TOE did not offer to use an authentication mechanism supported by the TOE.

2.2.11 FCS_TLSC_EXT.1 TLS Client Protocol

2.2.11.1 TSS Activities

FCS_TLSC_EXT.1.1

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 include those listed for this component.

Section 6.2 of [ST] lists the supported TLS cipher suites for the TOE’s TLS client implementation, which are consistent with those that are claimed in the SFR.

FCS_TLSC_EXT.1.2

The evaluator shall ensure that the TSS describes the client’s method of establishing all reference identifiers from the administrator/application-configured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.

Note that where a TLS channel is being used between components of a distributed TOE for FPT_ITT.1, the requirements to have the reference identifier established by the administrator are relaxed and the identifier may also be established through a “Gatekeeper” discovery process. The TSS shall describe the discovery process and highlight how the reference identifier is supplied to the “joining” component. Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1 and the ST author selected attributes from RFC 5280, the evaluator shall ensure the TSS describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The evaluator shall ensure the TSS presents an argument how the attribute type, or combination of attribute types, uniquely identify the remote TOE component; and the evaluator shall verify the attribute type, or combination of attribute types, is sufficient to support unique identification of the maximum supported number of TOE components.

Section 6.2 of [ST] states that the TOE compares the external server's presented identifier to the reference identifier by matching the certificate Common Name (Subject), FQDN (hostname), and prioritizing the SAN field (if exists) over the CN field. The TOE supports FQDN reference identifier and wildcards for peer authentication; IP addresses are not used in the evaluated configuration for the syslog connection.

The second paragraph of this evaluation activity is N/A to this evaluation because it only applies to distributed TOEs.

FCS_TLSC_EXT.1.2

If IP addresses are supported in the CN as reference identifiers, the evaluator shall ensure that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order. The evaluator shall also ensure that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced.

IP addresses are not supported as reference identifiers.

FCS_TLSC_EXT.1.3

None.

N/A

FCS_TLSC_EXT.1.4

If "present the Supported Groups Extension" is selected, the evaluator shall verify that TSS describes the Supported Groups Extension and whether the required behaviour is performed by default or may be configured. If TLS 1.2 is claimed and DHE ciphers are claimed, then the TSS must also specify whether the TOE is capable of negotiating DHE ciphers and whether the TOE client will terminate if an unsupported DHE parameter set is returned in the Server Key Exchange or whether all valid server-generated DHE parameters are accepted.

Section 6.2 of [ST] states that the TOE presents the Supported Elliptic Curves Extension in the Client Hello with the secp256r1, secp384r1, and secp521r1 NIST curves and is enabled by default.

FCS_TLSC_EXT.1.5

[Conditional]: The evaluator shall verify that TSS describes the signature_algorithms extension and whether the required behavior is performed by default or may be configured.

[Conditional]: The evaluator shall verify that TSS describes the signature_algorithms_cert extension and whether the required behavior is performed by default or may be configured.

[ST] section 6.2 states that the supported signature algorithms is configured as part of entering the TOE into FIPS-CC mode and that the required behavior is subsequently performed by default.

FCS_TLSC_EXT.1.6
The evaluator shall verify that TSS describes whether the list of supported ciphersuites can be configured or not.

[ST] section 6.2 states that the TSF does not support configuration of allowed ciphersuites.

FCS_TLSC_EXT.1.7
None

N/A

FCS_TLSC_EXT.1.8
The evaluator shall verify in the TSS that, for TLS 1.3, the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

This is not applicable as the TOE does not use TLS 1.3 or PSKs.

FCS_TLSC_EXT.1.9
None

N/A

2.2.11.2 Guidance Activities

FCS_TLSC_EXT.1.1
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.

Section 6.2 of [CCECG] describes how to enable FIPS-CC mode and states that this is sufficient to ensure that the supported TLS versions and cipher suites are limited to those claimed in [ST].

FCS_TLSC_EXT.1.2
The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator shall ensure that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.

Section 6.2 of [CCECG] states the TOE supports checking of FQDN identifiers as specified in RFC 6125. Section 6.8.1 of [CCECG] provides detailed instructions on how to configure the reference identifier used to check the identity of an external syslog server. Section 6.8.2 of [CCECG] provides detailed instructions on how to configure the reference identifier used to check the identity of an external firewall.

FCS_TLSC_EXT.1.2

Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1, the SFR selects attributes from RFC 5280, and FCO_CPC_EXT.1.2 selects "no channel"; the evaluator shall verify the guidance provides instructions for establishing unique reference identifiers based on RFC 5280 attributes.

The TOE is not distributed so this evaluation activity is not applicable.

FCS_TLSC_EXT.1.3

None

N/A

FCS_TLSC_EXT.1.4

If the TSS indicates that the Supported Groups Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Groups Extension.

[ST] does not indicate that the Supported Elliptic Curves Extension must be configured to meet the requirement; enabling FIPS-CC mode is sufficient to ensure that this function behaves as claimed in [ST].

FCS_TLSC_EXT.1.5

If the TSS indicates that the signature_algorithms extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the signature_algorithms extension.

[ST] states that this behavior is enforced by default after FIPS-CC mode is enabled so no separate guidance is needed ([CCECG] already instructs the reader to place the TOE into its evaluated configuration by enabling FIPS-CC mode).

FCS_TLSC_EXT.1.6

If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall verify that AGD guidance includes configuration of the list of supported ciphersuites.

This is not applicable as there is no way to configure the list of supported ciphersuites in the evaluated configuration of the TOE.

FCS_TLSC_EXT.1.7

None

N/A

FCS_TLSC_EXT.1.8

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

The ST indicates that the TOE does not use TLS 1.3 or PSKs, therefore this is not applicable.

FCS_TLSC_EXT.1.9

None

N/A

2.2.11.3 Test Activities

FCS_TLSC_EXT.1.1

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).

The evaluator caused the TOE to initiate connections to a TLS server using each of the cipher suites claimed in the Security Target and confirmed that each connection succeeded.

FCS_TLSC_EXT.1.1

The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation.

Test 2: The evaluator shall establish the connection with a server presenting a certificate that contains the serverAuth (OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage extension and verify that the connection successfully negotiated. The evaluator shall then verify that when the same server presents an otherwise valid server certificate that contains the extendedKeyUsage extension without serverAuth the client rejects the connection. Ideally, the two certificates should be identical except for the OID values.

The evaluator confirmed that a TLS server presenting a certificate with the Server Authentication purpose in the extendedKeyUsage field resulted in a successful connection. The evaluator then configured the TLS server with a certificate without the Server Authentication purpose in the extendedKeyUsage field and attempted a connection and verified that the TOE rejected the connection.

FCS_TLSC_EXT.1.1

Test 3: [conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send an ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite). The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.

The evaluator configured the TLS server to present an RSA server certificate with an ECDSA ciphersuite and attempted a connection from the TOE. The evaluator confirmed that the TOE rejected the connection.

FCS_TLSC_EXT.1.1

Test 4: The evaluator shall perform the following 'negative tests':

i. [conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the TOE TLS client denies the connection.

ii. Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite (compatible with the server-selected version of TLS) not presented in the Client Hello handshake message. The evaluator shall verify that the TOE TLS client rejects the connection after receiving the Server Hello.

iii. The evaluator shall attempt to establish a TLS connection using each valid TLS/SSL version (i.e. TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0, SSL 3.0, SSL 2.0). The evaluator shall verify that the version(s) specified in FCS_TLSC_EXT.1.1 are successfully established and all other versions are rejected by the TOE TLS client. If a supported_versions extension is not sent by the TOE in the ClientHello, then the evaluator must ensure the test server responds with a ServerHello that is valid for the TLS version being negotiated. If the TOE includes the Supported Versions extension in its ClientHello, the evaluator shall also ensure the version(s) specified in the extension match the version(s) in FCS_TLSC_EXT.1.1. NOTE: For TLS 1.3 aware test servers, it is appropriate for the test server to issue a TLS Alert. The TOE client must not attempt to continue the connection.

For part i: The evaluator configured the TLS server to present the TLS_NULL_WITH_NULL_NULL ciphersuite in the Server Hello message and attempted a connection from the TOE. The evaluator confirmed that the TOE rejected the connection.

For part ii: the evaluator configured the TLS server to present a ciphersuite which is not present in the ClientHello message in the ServerHello message and attempted a connection from the TOE. The evaluator confirmed the TOE rejected the connection.

For part iii: The evaluator verified the TOE rejected connections when the server attempted to select any non-claimed protocol versions and only accepted connections when a claimed protocol version is selected by the server.

Test 5: The evaluator shall perform the following modifications to the traffic (i.e. Man-in-the-middle modifications that result in invalid signatures and MACs):

i. [conditional]: Perform this test only if support of TLS 1.2 is claimed. If using DHE or ECDH ciphersuites, modify the signature block in the Server's Key Exchange handshake message, and verify that the client denies the connection and no application data flows. The handshake shall be valid (e.g. the Finished message is calculated using the modified signature), with the exception of the invalid signature. This test does not apply to ciphersuites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.

ii. [conditional]: Perform this test only if support of TLS 1.3 is claimed. Modify the signature block in the Server's Certificate Verify handshake message, and verify that the client denies the connection and no application data flows. The handshake shall be valid (e.g. the Finished message is calculated using the modified signature), with the exception of the invalid signature.

For part i: The evaluator configured a TLS server to modify the signature of the TLS server's ServerKeyExchange record and verified the TOE rejected the connection.

For part ii: This is not applicable as the TOE does not claim the use of TLS 1.3.

Test 6: The evaluator shall perform the following 'scrambled message tests':

i. Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application data flows. (Note: This modification must be performed prior to the contents of the Finished message being encrypted.)

ii. [conditional]: Perform this test only if support of TLS 1.2 is claimed. Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake is not finished successfully and no application data flows. (Note: TLS 1.3 provides for a dummy ChangeCipherSpec message to aid in middlebox compatibility if such an option is enabled in the specific implementation [see Section D.4 in RFC 8446]. If TLS 1.3 middlebox compatibility mode is enabled a ChangeCipherSpec message may appear in packet traces, but it does not influence the protocol. To be clear: for TLS 1.3, this test does not need to be performed.)

iii. [conditional]: Perform this test only if support of TLS 1.3 is claimed. Send a plaintext EncryptedExtensions message from the server and verify that the handshake is not finished successfully and no application data flows. (Note: Under TLS 1.3, the EncryptedExtensions message is the first message to be encrypted with the handshake traffic secret.)

iv. [conditional]: Perform this test only if support of TLS 1.2 is claimed. Modify at least one byte in the server's nonce in the Server Hello handshake message and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.

For part i: The evaluator modified a byte in the Server Finished record before encrypting and sending it to the client and observed the TOE terminating the connection after receiving the Server Finished message.

For part ii: The evaluator configured a TLS server to send a garbled application data message instead of a Finished record after the ChangeCipherSpec message and confirmed the TOE rejected the connection.

For part iii: This is not applicable as the TOE does not claim the use of TLS 1.3.

For part iv: The evaluator configured a TLS server to modify the locally stored value of the nonce after sending the ServerHello and verified the TOE rejected the ServerKeyExchange record as the TOE used a different nonce for the ServerKeyExchange signature calculation.

FCS_TLSC_EXT.1.2

Note that the following tests are marked conditional and are applicable under the following conditions:

a. For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.

or

b. For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable

or

c. For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.

Note that for some tests additional conditions apply.

IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:

- IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.
- IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.

The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:

Test 1 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN.

Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1.

The evaluator configured the TLS server to present a certificate with a CN that does match the reference identifier and no SAN extension. The evaluator attempted a connection from the TOE and verified that the TOE did not accept the connection. For this test, the evaluator generated a certificate with an invalid FQDN.

FCS_TLSC_EXT.1.2

Test 2 [conditional]: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.

The evaluator configured the TLS server to present a certificate with a CN that matched the reference identifier and a SAN extension with a value that did not match the reference identifier. The evaluator attempted a connection from the TOE and verified that the TOE did not accept the connection. The evaluator tested a bad FQDN in the SAN.

FCS_TLSC_EXT.1.2

Test 3 [conditional]: If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.

The evaluator configured the TLS server to present a certificate with a valid CN and no SAN extension. The evaluator attempted a connection from the TOE and verified that the TOE accepted the connection. The evaluator tested FQDN identifiers in the CN field.

FCS_TLSC_EXT.1.2

Test 4 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).

The evaluator configured the TLS server to present a certificate with a CN that does not match the reference identifier and a SAN extension with a value that does match the reference identifier. The evaluator attempted a connection from the TOE and verified that the TOE accepted the connection. The evaluator tested FQDN identifiers in the SAN extension.

FCS_TLSC_EXT.1.2

Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID):

i. [conditional]: The evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails.

ii. [conditional]: The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds, if wildcards are supported, or fails if wildcards are not supported. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)

Each of these cases were tested for using the label in both the CN and the SAN.

The evaluator configured the TLS server to present a certificate with a DNS value of test.*.leidos.ate as the certificate identifier. The evaluator attempted a connection from the TOE and verified that the TOE did not accept the connection.

The evaluator configured the TLS server to present a certificate with a DNS value of *.leidos.ate and configured the TOE with a reference identifier of tlss.leidos.ate. The evaluator attempted a connection from the TOE and verified that the TOE accepted the connection. The evaluator then configured the TOE to use a reference identifier with two left-most labels (test.tlss.leidos.ate) independently the evaluator configured the TOE to use a reference identifier with no left-most label (tlss.ate) and presented a certificate with a DNS value of *.tlss.ate, and attempted a connection for each. The evaluator confirmed that the TOE did not accept either connection.

FCS_TLSC_EXT.1.2

Test 6: Objective: The objective of this test is to ensure the TOE is able to differentiate between IP address identifiers that are not allowed to contain wildcards and other types of identifiers that may contain wildcards.

[conditional]: If IP address identifiers are supported in the SAN or CN, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with a wildcard asterisk (*) (e.g. CN=*.168.0.1 when connecting to 192.168.0.1, CN=2001:0DB8:0000:0000:0008:0800:200C:* when connecting to 2001:0DB8:0000:0000:0008:0800:200C:417A). The certificate shall not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported IP address version (e.g. IPv4, IPv6).

This test is not applicable as the TOE does not utilize IP reference identifiers.

FCS_TLSC_EXT.1.2

Test 7: [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator shall modify each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier):

i. The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails.

ii. The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-at-serialNumber=correct_identifier, the certificate could instead include id-at-name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass this test.

iii. The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.

iv. The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)

This test is not applicable because the TOE is not distributed and does not claim FPT_ITT.1.

FCS_TLSC_EXT.1.3

The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

Test 1: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.

The evaluator verified the TOE accepts connections when the server presents a valid certificate that chains to a CA trusted by the TOE.

FCS_TLSC_EXT.1.3

The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

Test 2 [conditional]: If "except with the following administrator override" is selected, the evaluator shall change the presented certificate(s) or modify the operational environment, so that certificate validation fails due to the TSF’s inability to determine revocation status. The evaluator shall verify that the certificate is not accepted by the TSF until the Security Administrator authorizes the TSF to establish the connection and this action results in the Trusted Channel being successfully established.

This test is not applicable as the ST does not make the “except with the following administrator override” selection.

FCS_TLSC_EXT.1.3

The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

Test 3: While performing testing of invalid TLS Client Reference Identifiers, expired X.509 certificates, and invalid X.509 trust chains; the evaluator shall ensure the TSF does not present an administrator override option, with the expcetion of failure to determine revocation status (if selected). Note: This should be a review of behavor observed while performing other tests.

The evaluator observed no ability to override the certificate validation failures during the testing of the various certificate validation failures was provided to the administrator.

FCS_TLSC_EXT.1.4

Test 1 [conditional]: If “not present the Supported Groups Extension” is selected, the evaluator shall examine the Client Hello message and verify it does not contain the Supported Groups extension.

This test is not applicable as the TOE makes the selection to present the Supported Groups Extension.

FCS_TLSC_EXT.1.4

Test 2 [conditional]: If “present the Supported Groups Extension” is selected, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) key exchange using each of the TOE’s supported groups. The evaluator shall verify that the connection succeeds. This test shall be repeated for each type of key exchange message/extension supported (i.e. Key Share extension for TLS 1.3 and Server Key Exchange Message for TLS 1.2).

The evaluator verified that the TOE could successfully establish a TLS 1.2 client connection using each of the specified ECDHE curve values for the key exchange algorithm.

FCS_TLSC_EXT.1.4

Test 3 [conditional]: If secp curves are selected, the evaluator shall configure the server to perform an ECDHE key exchange in the TLS connection using a non-supported curve and shall verify that the connection fails and no application data flows. The non-supported curve shall be as similar to the selected curve(s) as possible (i.e. a non-selected curve when not all curves are selected or P-224). This test shall be repeated for each type of key exchange message/extension supported (i.e. Key Share extension for TLS 1.3 and Server Key Exchange Message for TLS 1.2).

The evaluator configured the TLS server to select a key exchange that is not supported by the (secp192r1) and verified that the TOE rejected the connection.

FCS_TLSC_EXT.1.4

Test 4a [conditional, for TLS 1.3 only]: If ffdhe curves are selected, the evaluator shall configure the server to perform a DHE key exchange in the TLS connection using a non-supported group and shall verify that the connection fails and no application data flows. The non-supported group shall be as similar to the selected group(s) as possible (i.e. a non-selected group when not all groups are selected or undefined Codepoint 0x0105 (ffdhe8192 + 1)).

This test is not applicable as TLS 1.3 is not claimed.

FCS_TLSC_EXT.1.4

Test 4b [conditional, for TLS 1.2 only]: If ffdhe curves are selected, the evaluator shall configure the server to return DHE parameters in the Server Key Exchange in the TLS connection that do not meet the construction for any claimed ffdhe group. The evaluator shall verify that the connection fails and no application data flows. If the TOE client supports any server-returned DHE parameter set, then this test is not applicable.

This test is not applicable as FFDHE curves are not selected in the ST.

FCS_TLSC_EXT.1.5

Test 1 [conditional]: The evaluator shall perform the following tests if "present the signature_algorithms extension" is selected:

i. The evaluator shall examine the Client Hello message and verify it contains the signature_algorithms extension and the SignatureSchemes match the SignatureSchemes specified in the requirement.
ii. The evaluator shall establish a TLS connection using each of the SignatureSchemes specified by the requirement and observes the session is successfully completed. The evaluator shall ensure the test server sends a leaf Certificate that has a public key algorithm that is consistent with the SignatureScheme being tested. For TLS 1.2 and if the ciphersuite is DHE or ECDHE, the evaluator shall ensure that the server sends Server Key Exchange messages consistent with the SignatureScheme being tested. For TLS 1.3, the evaluator shall ensure that the server sends Certificate Verify messages consistent with the SignatureScheme being tested.

For part i: The evaluator examined the ClientHello sent by the TOE and verified that the SignatureAlgorithms offered by the TOE is consistent with the claim in the ST.

For part ii: The evaluator verified the TOE could successfully complete a connection using each of the claimed signature algorithms as the algorithm for the ServerkeyExchange (TLS 1.2).

FCS_TLSC_EXT.1.5

Test 2 [conditional]: The evaluator shall perform the following tests if "present the signature_algorithms_cert extension" is selected:

i. The evaluator shall examine the Client Hello message and verify it contains the signature_algorithms_cert extension and the SignatureSchemes match the SignatureSchemes specified in the requirement.

ii. The evaluator shall establish a TLS connection using a certificate chain using each of the SignatureSchemes specified by the requirement. The evaluator shall ensure the signatures used in the certificate chain are consistent with the SignatureScheme being tested.

This test is not applicable as the ST does not make the “present the signature_algorithms_cert extension” selection.

FCS_TLSC_EXT.1.6

[conditional]: If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall establish a TLS connection using one of the possible configurations of the list of supported ciphersuites. The evaluator shall then change the configuration and repeat the test. The evaluator shall verify that the behavior of the TOE has changed according to the modification of the list of ciphers. This test shall be repeated for all supported TLS versions. If the TSF does not provide the ability of configuring the list of supported ciphersuites, this test shall be omitted.

This test is not applicable as the TOE does not provide the ability to configure the list of supported ciphersuites.

FCS_TLSC_EXT.1.7

The evaluator shall establish a TLS connection with a server and observe that the early data extension and the post-handshake client authentication extension according to RFC 8446 Section 4.2 are not advertised in the Client Hello Message. This test shall be executed for all TLS versions supported by the TOE.

The evaluator verified the TOE does not send the EarlyData or PostHandshake ClientAuthentication extensions in the ClientHello message.

FCS_TLSC_EXT.1.8

None

N/A

FCS_TLSC_EXT.1.9

Test 1 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall use a network packet analyzer/sniffer to capture a TLS 1.2 handshake between the two TLS endpoints. The evaluator shall verify that either the “renegotiation_info” field or the SCSV ciphersuite is included in the ClientHello message during the initial handshake.

This test is not applicable as the ST does not make the “support TLS 1.2 secure renegotiation” selection.

FCS_TLSC_EXT.1.9

Test 2 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and verify the TOE TLS Client’s handling of ServerHello messages received during the initial handshake that include the “renegotiation_info” extension. The evaluator shall modify the length portion of this field in the ServerHello message to be non-zero and verify that the TOE TLS client sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.

This test is not applicable as the ST does not make the “support TLS 1.2 secure renegotiation” selection.

FCS_TLSC_EXT.1.9

Test 3 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and verify that ServerHello messages received during secure renegotiation contain the “renegotiation_info” extension. The evaluator shall modify either the “client_verify_data” or “server_verify_data” value and verify that the TOE TLS client terminates the connection.

This test is not applicable as the ST does not make the “support TLS 1.2 secure renegotiation” selection.

Modified by TD0899

FCS_TLSC_EXT.1.9

Test 4a [conditional, if the TOE supports TLS 1.3]: ***The evaluator shall initiate a TLS session between the TSF and a test TLS 1.3 server that completes a compliant TLS 1.3 handshake, followed by a hello request message. The evaluator shall observe that the TSF completes the initial TLS 1.3 handshake successfully, but terminates the session on receiving the hello request message.***

It is preferred that the TSF sends a fatal error alert message (e.g., unexpected message) in response to this, but it is acceptable that the TSF terminates the connection silently (i.e., without sending a fatal error alert).

Test 4***b*** [conditional]: ***if the TOE supports TLS 1.2*** and reject***s TLS 1.2*** …renegotiation attemp***t***s~~" is selected, then for each selected TLS version, t]~~: ***T***he evaluator shall initiate a TLS session between the so-configured TSF and a test ***TLS 1.2*** server that is configured to perform a compliant handshake, followed by a hello ~~reset~~ request. The evaluator shall confirm that the TSF completes the initial handshake successfully ~~but terminates the TLS session~~ ***does not initiate renegotiation*** after receiving the hello ~~reset~~ request. ~~Note: It is preferred that the TSF sends a fatal error alert message (e.g., unexpected message) in response to this, but it is acceptable that the TSF terminates the connection silently (i.e., without sending a fatal error alert).~~

For TLS 1.2, the evaluator verified the TOE rejects the HelloRequest message sent by the server in accordance with RFC 5246 section 7.4.1.1. Support for TLS 1.3 is not claimed.

2.2.12 FCS_TLSS_EXT.1 TLS Server Protocol

2.2.12.1 TSS Activities

FCS_TLSS_EXT.1.1

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.

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

Section 6.2 of [ST] lists the TLS ciphersuites that are supported by the TOE when it acts as a TLS server. This list is consistent with what is claimed in FCS_TLSS_EXT.1.1.

The ciphersuites specified in the SFR are identical to those listed in the TSS.

The TOE denies connections from clients requesting connections using versions other than TLS 1.2. This rejection is performed by default.

FCS_TLSS_EXT.1.2

The evaluator shall verify that the TSS describes the algorithms and key sizes the TSF supports for authenticating itself to TLS clients. The evaluator shall ensure these algorithms are consistent with the selected ciphersuites.

[ST] section 6.2 states that the TSF supports RSA 2048/3072/4096 and ECDSA secp256r1/secp384r1/secp521r1, consistent with the SFR definition.

FCS_TLSS_EXT.1.3

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. The evaluator shall ensure these algorithms are consistent with the selected ciphersuites.

Section 6.2 of [ST] states that the key agreement parameters of the server key exchange message consist of the key establishment parameters generated by the TOE: Diffie-Hellman parameters with key size of MODP group 2048-bit, ECDHE implementing NIST curves secp256r1, secp384r1, and secp521r1.

FCS_TLSS_EXT.1.4

The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246), if session resumption based on session tickets is supported (RFC 5077) and/or if session resumption according to RFC 8446 is supported.

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.

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.

[ST] section 6.2 states the TOE supports session resumption using tickets for a single context (no configuration needed).

FCS_TLSS_EXT.1.4

If the TOE claims a TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator shall verify 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.

[ST] section 6.2 states the TOE supports session resumption using tickets for a single context only (no configuration required). The TOE checks if session tickets expire, which would trigger a full handshake. The session tickets are encrypted with AES encryption and 128-bits encryption key plus 256-bits HMAC-SHA-256 key and adhere to the structural format provided in section 4 of RFC 5077.

FCS_TLSS_EXT.1.5

The evaluator shall verify that TSS describes whether the list of supported ciphersuites can be configured or not.

The SFR states that the TSF does not provide the ability to configure the list of supported ciphersuites as defined in FCS_TLSS_EXT.1.1. [ST] section 6.2 affirms that TLS ciphersuite configuration is not supported.

FCS_TLSS_EXT.1.6

None

N/A

FCS_TLSS_EXT.1.7

The evaluator shall verify in the TSS that, for TLS 1.3, the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

N/A – the TOE does not claim TLS 1.3.

FCS_TLSS_EXT.1.8

None

N/A

2.2.12.2 Guidance Activities

FCS_TLSS_EXT.1.1

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 or TLS version supported by the TOE may have to be restricted to meet the requirements).

Section 6.2 of [CCECG] describes how to enable FIPS-CC mode and states that this is sufficient to ensure that the supported TLS versions and cipher suites are limited to those claimed in [ST]. If further restrictions are desired, section 6.8.3 of [CCECG] offers instructions on how to further restrict the TLS cipher suites offered by the TOE's TLS server to ECDHE only.

FCS_TLSS_EXT.1.2

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

Section 6.2 of [CCECG] describes how to enable FIPS-CC mode and states that this is sufficient to ensure that the supported TLS versions and cipher suites are limited to those claimed in [ST].

FCS_TLSS_EXT.1.3

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

Section 6.2 of [CCECG] describes how to enable FIPS-CC mode and states that this is sufficient to ensure that the supported TLS versions and cipher suites are limited to those claimed in [ST]. No separate configuration for this is required.

FCS_TLSS_EXT.1.4

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

Section 6.2 of [ST] states the TOE's support of session resumption does not require any configuration.

FCS_TLSS_EXT.1.5

If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall verify that AGD guidance includes configuration of the list of supported ciphersuites.

This is not applicable as the TOE does not provide the ability to configure the list of supported ciphersuites.

FCS_TLSS_EXT.1.6

None

N/A

FCS_TLSS_EXT.1.7
The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

By default the TOE does not support PSKs and this is not configurable, so no configuration is necessary and therefore is not documented in the operational guidance.

FCS_TLSS_EXT.1.8
None

N/A

2.2.12.3 Test Activities

FCS_TLSS_EXT.1.1
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).

The evaluator configured a TLS client to request each of the ciphersuites claimed in the Security Target and attempted a connection to the TOE. The evaluator confirmed that negotiation of each claimed ciphersuite was successful and the connection was successfully established.

FCS_TLSS_EXT.1.1
Test 2: The evaluator shall perform the following tests:

i. 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.

ii. [conditional]: Perform this test only if support of TLS 1.2 is claimed. 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.

For part i: The evaluator configured a TLS client to only offer ciphersuites in the ClientHello which are not claimed by the TOE and verified the TOE rejected the connection.

For part ii: The evaluator configured a TLS client to only offer the TLS_NULL_WITH_NULL_NULL ciphersuite in its ClientHello and verified the TOE rejected the connection.

FCS_TLSS_EXT.1.1

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

i. [conditional]: Perform this test only if support of TLS 1.2 is claimed. Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data.

   (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 TLS Finished message and b) Encrypt every TLS message after session keys are negotiated.)

ii. [conditional]: Perform this test only if support of TLS 1.2 is claimed. 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'.

iii. [conditional]: Perform this test only if support of TLS 1.3 is claimed. The evaluator shall use a client to send a Client Hello message containing a single curve in the Supported Groups extension. The curve that is selected to be presented in this extension should not be supported by the TOE. The evaluator shall verify that the TOE disconnects after receiving the Client Hello message.

iv. [conditional]: Perform this test only if support of TLS 1.3 is claimed. The evaluator shall use a client to send a Client Hello message containing multiple curves in the Supported Groups extension. These curves should be chosen such that only one of these curves is supported by the TOE. The evaluator shall verify that the TOE responds with a Hello Retry Request message selecting the supported curve. This shall be reflected in the Key Share extension of the Hello Retry Request message.

For part i: The evaluator configured a TLS client to modify the Client Finished handshake message. The evaluator observed the TOE rejected the connection and did not transmit any application data.

For part ii: The evaluator examined a TLS finished record sent by the TOE to the client and verified that the Finished record was truly encrypted and not readable in plaintext by the evaluator.

For part iii: This test is not applicable as the TOE does not claim TLS 1.3 support for TLS server functionality.

For part iv: This test is not applicable as the TOE does not claim TLS 1.3 support for TLS server functionality.

FCS_TLSS_EXT.1.1

Test 4: The evaluator shall attempt to establish a TLS/SSL connection using each of the supported TLS/SSL versions (i.e., TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0, SSL 3.0, SSL 2.0). The client shall be configured so it only supports the version being tested. The evaluator shall verify that the versions specified in FCS_TLSS_EXT.1.1 are successfully established and all other versions not successfully established. If the TOE attempts to downgrade the version, it is acceptable for the test client to terminate the connection; however, the version selected by the TOE shall always be a version specified in FCS_TLSS_EXT.1.1.

The evaluator verified the TOE did not accept connections from clients attempting to negotiate a protocol version which is not supported by the TOE (TLS 1.3, TLS 1.1, TLS 1.0, SSL 3.0, SSL 2.0) and only accepts connections from clients attempting to negotiate the supported protocol version (TLS 1.2).

FCS_TLSS_EXT.1.2 and FCS_TLSS_EXT.1.3

Test 1 [conditional]: If ECDHE ciphersuites/group are supported:

The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite (TLS 1.2) or group (TLS 1.3) and a single supported elliptic curve specified in the supported groups extension. The Evaluator shall verify (through a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange (TLS 1.2) or Server Hello (key_share, for TLS 1.3) message and successfully establishes the connection.

For TLS 1.2, the evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g., secp192r1 (0x13)) specified in RFC 4492, Section 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

For TLS 1.3, the evaluator shall attempt a connection using a supported ciphersuite and a single unsupported group. Both the key_share and supported_groups extensions must be set to the same unsupported group. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

The evaluator configured a TLS client to specify only one curve supported by the TOE in the Elliptic Curves extension. The evaluator observed that the TOE selected the same curve and established a connection successfully. This was repeated for each claimed curve secp256r1, secp384r1, secp521r1.

The evaluator configured a TLS client to specify a curve not supported by the TOE in the Elliptic Curves extension (specifically, secp192r1). The evaluator verified the TOE did not establish a connection or return a Server Hello record.

FCS_TLSS_EXT.1.2 and FCS_TLSS_EXT.1.3

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.

For TLS 1.2, 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).

For TLS 1.3, the evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Share Extension Message where the KeyShareServerHello structure contains a KeyShareEntry structure with an opaque key_exchange value whose Length is consistent with the configured Diffie-Hellman parameter size(s).

The TOE supports DHE parameters of 2048 bits only. The evaluator connected to the TOE using a TLS client tool. The evaluator observed that the p length in the TOE’s Server Key Exchange message was set to 256 bytes or 2048 bits.

FCS_TLSS_EXT.1.2 and FCS_TLSS_EXT.1.3

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

This test is not applicable as the TOE does not utilize RSA key establishment for TLS connections.

FCS_TLSS_EXT.1.4

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

Test 1 [conditional]: If the TOE does not support session resumption based on session IDs according to RFC 5246 (TLS 1.2) or session tickets according to RFC 5077 (TLS 1.2) or session resumption according to RFC 8446 (TLS 1.3), the evaluator shall perform the following test:

i. For all supported TLS versions the client shall send a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket. A non-zero length session identifier for TLS 1.3 would result in testing compatibility mode which is not the objective of this test. For TLS 1.3, the evaluator shall ensure that a 'psk_key_exchange_modes' extension is included in the Client Hello.

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

iii. The client verifies the Server Hello message contains a zerolength session identifier. For TLS 1.2 the client could alternatively pass the following steps (not applicable for TLS 1.3):

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

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

v. 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).

vi. 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.

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.

The TOE supports session resumption based on session tickets. As such, this test is not applicable.

FCS_TLSS_EXT.1.4

Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC 5246 (TLS 1.2), 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):

i. 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). When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.

ii. 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.

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.

This test is not applicable as the TOE does not support resumption based on RFC 5246 and only supports RFC 5077 resumption.

FCS_TLSS_EXT.1.4

Test 3 [conditional]: If the TOE supports session tickets according to RFC 5077 (supported only by TLS 1.2), the evaluator shall carry out the following steps:

i. 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. When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.

ii. 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 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.

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.

The evaluator configured a TLS client to support TLS session tickets. The evaluator observed that the TOE returned a session ticket. The evaluator observed that when the TLS client presented the session ticket to the TOE that the TOE attempted to resume the session using the abbreviated handshake.

The evaluator configured a TLS client to support TLS session tickets. The evaluator observed that the TOE returned a session ticket. The evaluator observed that when the TLS client presented a modified session ticket to the TOE that the TOE rejected the session ticket and terminated the connection.

FCS_TLSS_EXT.1.4

Test 4 [conditional]: If the TOE supports session resumption according to RFC 8446 (supported only by TLS 1.3), the evaluator shall carry out the following steps:

i. 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 pre-shared key in the ClientHello. The evaluator shall confirm that the TOE responds similarly to figure 3 of RFC 8446 after successfully reusing the pre-shared-key to resume the session. Specifically, the server must not send back a Certificate message if the session is correctly resumed. When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.

ii. 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 modify the pre-shared key 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, or (2) terminates the connection in some way that prevents the flow of application data.

iii. 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 force the non-TOE client to attempt to establish a new connection using the previous session ticket material as a pre-shared key, but set psk_key_exchange_modes with a value of psk_ke in the Client Hello message and omit the psk_ke_dhe. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake, or (2) terminates the connection in some way that prevents the flow of application data.

This test is not applicable as the TOE does not support session resumption based on RFC 8446.

FCS_TLSS_EXT.1.5

Test 1[conditional]: If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall establish a TLS connection using one of the possible configurations of the list of supported ciphersuites. The evaluator shall then change the configuration and repeat the test. The evaluator shall verify that the behavior of the TOE has changed according to the modification of the list of ciphers. This test shall be repeated for all supported TLS versions. If the TSF does not provide the ability of configuring the list of supported ciphersuites, this test shall be omitted.

The TOE does not claim to support configuring the list of supported ciphersuites.

FCS_TLSS_EXT.1.6

According to RFC 8446 Section 4.2.10, a PSK is required to use the early data extension. As NDcPP only allows the use of PSK in conjunction with session resumption, a NDcPP conformant TOE which acts as TLS Server cannot use the early data extension if session resumption is not supported. For TOEs that do not support session resumption, execution of test FCS_TLSS_EXT.1.4 Test 1 is regarded as sufficient that the TOE does not support the early data extension. For TOEs that support session resumption, FCS_TLSS_EXT.1.4 Test 2(i), 3(i) or 4(i) (depending on the supported TLS versions and the way session resumption is implemented) ensure that the TOE does not support the early data extension.

This test is not applicable as this is specific to functionality provided by RFC 8446 which is only applicable to TLS 1.3 and the TOE does not utilize TLS 1.3 for TLS server functionality.

FCS_TLSS_EXT.1.7

None

N/A

FCS_TLSS_EXT.1.8

Test 1 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall use a network packet analyzer/sniffer to capture a TLS 1.2 handshake between the two TLS endpoints. The evaluator shall verify that the “renegotiation_info” extension is included in the ServerHello message.

This test is not applicable as the ST does not make the “support secure renegotiation” selection.

FCS_TLSS_EXT.1.8

Test 2 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and modify the length portion of the field in the ClientHello message in the initial handshake to be non-zero. The evaluator shall verify that the TOE TLS server sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.

This test is not applicable as the ST does not make the “support secure renegotiation” selection.

FCS_TLSS_EXT.1.8

Test 3 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and modify the "client_verify_data" or "server_verify_data" value in the ClientHello message received during secure renegotiation. The evaluator shall verify that the TOE TLS server terminates the connection.

This test is not applicable as the ST does not make the “support secure renegotiation” selection.

Modified by TD0899

FCS_TLSS_EXT.1.8

Test 4a [conditional, if the TOE supports TLS 1.3]: The evaluator shall follow the operational guidance as necessary to configure the TSF to negotiate TLS 1.3. The evaluator shall initiate a valid initial TLS 1.3 session, send a valid client hello on the non-renegotiable TLS channel, and observe that the TSF terminates the session.

Note: It is preferred that the TSF sends a fatal error alert message (e.g., unexpected message) in response to this, but it is acceptable that the TSF terminates the connection silently (i.e., without sending a fatal error alert).

Test 4***b*** [conditional***, if the TOE supports TLS 1.2 and***~~]: If "~~reject***s TLS 1.2*** …renegotiation attemp***t***s~~" is selected, then for each selected TLS version, t]~~***: The*** evaluator shall follow the operational guidance as necessary to configure the TSF to negotiate ~~the version~~ ***TLS 1.2*** and reject renegotiation. The evaluator shall initiate a valid initial ***TLS 1.2*** session ~~for the specified version~~, send a valid ClientHello on the non-renegotiable TLS channel, and observe that the TSF ***does not perform renegotiation of the TLS channel***. ~~terminates the session. Note: It is preferred that the TSF sends a fatal error alert message (e.g., unexpected message) in response to this, but it is acceptable that the TSF terminates the connection silently (i.e., without sending a fatal error alert).~~

The evaluator verified the TOE rejects TLS 1.2 renegotiation attempts. The TOE does not support TLS 1.3, therefore the TLS 1.3 portion of the test is not applicable.

2.2.13 FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication

2.2.13.1 TSS Activities

FCS_TLSS_EXT.2.1

The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.

[ST] section 6.3 states that the TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS connections (client and server authentication).

FCS_TLSS_EXT.2.1

The evaluator shall verify the TSS describes how the TSF uses certificates to authenticate the TLS client and how the TOE reacts in case either the client does not send a client certificate or the verification of the client certificate fails. The evaluator shall verify the TSS describes if the

TSF supports any fallback authentication functions (e.g. username/password, challenge response) the TSF uses to authenticate TLS clients that do not present a certificate. If fallback authentication functions are supported, the evaluator shall verify the TSS describes whether the fallback authentication functions can be disabled.

[ST] section 6.2 describes how the TOE's TLS server will authenticate a TLS client based on the presentation of a valid certificate. The TOE denies connections from clients requesting connections using TLS versions other than 1.2 and shall not establish a trusted channel if the fully qualified distinguished name (FQDN) in the subject or Subject Alternative Name (SAN) field contained in a certificate does not match the expected identifier for the peer. The TOE will match the FQDN identifier according to RFC 6125 section 6.

This section also states that a fallback mechanism is not supported for TLS, so the remaining portion of the evaluation activity is not applicable.

FCS_TLSS_EXT.2.2

None

N/A

FCS_TLSS_EXT.2.3

The evaluator shall verify that the TSS describes which types of identifiers are supported during client authentication (e.g. Fully Qualified Domain Name (FQDN)). If FQDNs are supported, the evaluator shall verify that the TSS describes that corresponding identifiers are matched according to RFC 6125. For all other types of identifiers, the evaluator shall verify that the TSS describes how these identifiers are parsed from the certificate, what the expected identifiers are and how the parsed identifiers from the certificate are matched against the expected identifiers.

[ST] section 6.2 states that FQDNs are supported for client certificate identifiers and that this is matched according to RFC 6125. No other identifier types are identified.

FCS_TLSS_EXT.2.4

The evaluator shall verify that TSS describes the use of the signature_algorithms extension and optionally the signature_algorithms_cert extension and whether the required behavior is performed by default or may be configured.

[ST] section 6.2 states that the signature algorithms extension is presented with the supported values specified in FCS_TLSS_EXT.2.4 by default (rsa_pkcs1 with sha256/384/512, ecdsa_secp256r1 with sha256, and ecdsa_secp384r1 with sha384).

2.2.13.2 Guidance Activities

FCS_TLSS_EXT.2.1

If the TSS indicates that mutual authentication using X.509v3 certificates is used, the evaluator shall verify that the AGD guidance includes instructions for configuring the client-side certificates for TLS mutual authentication.

Section 6.8.2 of [CCECG] includes instructions for configuring the client-side certificates for TLS mutual authentication used by firewalls.

FCS_TLSS_EXT.2.1

The evaluator shall verify the guidance describes how to configure the TLS client certificate authentication function. If the TSF supports fallback authentication functions, the evaluator shall verify the guidance provides instructions for configuring the fallback authentication functions. If fallback authentication functions can be disabled, the evaluator shall verify the guidance provides instructions for disabling the fallback authentication functions.

Section 6.2 of [ST] states a fallback mechanism is not supported for TLS.

FCS_TLSS_EXT.2.2

None

N/A

FCS_TLSS_EXT.2.3

The evaluator shall ensure that the AGD guidance describes the configuration of expected identifier(s) for X.509 certificate-based authentication of TLS clients. The evaluator shall ensure this description includes all types of identifiers described in the TSS and, if claimed, configuration of the TOE to use a directory server.

Section 6.8.2 of [CCECG] includes instructions for configuring expected identifiers for X.509 certificate-based authentication for firewall client devices.

Section 6.2 of [ST] states the TOE supports FQDNs for client certificate identifiers, which it matches according to RFC 6125.

FCS_TLSS_EXT.2.4

If the TSS indicates that the signature_algorithms extension (and/or the signature_algorithms_cert extension) must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the extension(s).

[ST] indicates this behavior is enforced by default so no configuration guidance is expected.

2.2.13.3 Test Activities

For all tests in this section the TLS client used for testing of the TOE shall support mutual authentication. For all tests that require a successful connection, the evaluator shall ensure the mutual authentication succeeds. As noted in Test 1b, seeing a TLS channel established and application data flowing, does not necessarily indicate successful TLS client authentication.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 1a [conditional]: If the TOE uses TLS Client authentication for FTP_ITC.1, FTP_ITT.1, or FTP_TRP.1/Join, the evaluator shall configure the TOE to send a Certificate Request message to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify that the handshake is not finished successfully and no application data flows.

The evaluator initiated a connection to the TOE using a TLS client and did not provide a client certificate for the mutual authentication of the client. The evaluator observed that the TOE returned an error that indicated the client needed to provide a certificate and verified that no application data flowed.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 1b [conditional]: If the TOE uses TLS Client authentication for FTP_TRP.1/Admin, the evaluator shall configure the TOE to send a Certificate Request message to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify that the client is not authenticated and no sensitive data flows. The method of verification may vary based on the TOE behavior:

- If the TOE's TLS implementation requires TLS Client Authentication, the evaluator shall verify the that the handshake is not finished successfully and no application data flows.
- If the TOE's TLS implementation does not require TLS Client Authentication and the TOE does not support fallback authentication, the evaluator shall verify the connection remains in an unauthenticated state and at most an error message and services specified in FIA_UIA_EXT.1 are available.
- If the TOE's TLS implementation does not require TLS Client Authentication and the TOE supports fallback authentication, the evaluator shall verify the TOE presents the fallback authentication mechanism as described in the TSS.

  Note: Testing the validity of the client certificate is performed as part of X.509 testing

This test is not applicable as the TOE does not use TLS client authentication for FTP_TRP.1/Admin.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 2: The intent of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To perform this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognised by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not terminate in the claimed CA certificate). The evaluator shall verify that the attempted connection is denied.

The evaluator created an “imposter” Intermediate CA certificate with a different private key than the “valid” Intermediate CA certificate but the same DN value. The evaluator used the “imposter” to sign an end-entity certificate and presented the end-entity certificate along with the “valid” Intermediate CA certificate to the TOE. The evaluator observed that the TOE checked the full chain and rejected the connection when the signature verification failed.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 3: The intent of this test is to verify certificate validation logic to ensure that only certificates with the correct OID present in the EKU are accepted.

The evaluator shall establish the connection with a client presenting a certificate with the ClientAuth (OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field and verify that the connection successfully negotiated. The evaluator shall then verify that when the same client presents an otherwise valid client certificate that contains the extendedKeyUsage extension without ClientAuth the server rejects the connection. Ideally, the two certificates should be identical except for the OID values.

The evaluator confirmed the TOE checks the presented client certificate for the presence of the ClientAuthentication EKU and only accepts certificates presented for mutual authentication which possess the ClientAuthentication EKU.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 4 [conditional]: The evaluator shall perform the following modifications to the traffic:

i. Perform this test only if support of TLS 1.2 is claimed. Configure the server to require mutual authentication and then connect to the server with a client configured to send a client certificate that is signed by a Certificate Authority trusted by the TOE. The evaluator shall verify that the server accepts the connection.

ii. Perform this test only if support of TLS 1.2 is claimed. Configure the server to require mutual authentication and then modify a byte in the signature block of the client’s Certificate Verify handshake message (see RFC 5246 Section 7.4.8). The evaluator shall verify that the server rejects the connection.

Note: Testing the validity of the client certificate is performed as part of X.509 testing.

For part i: The evaluator verified the TOE could completed a connection when requesting mutual authentication and the TLS client presents a valid certificate that chains to a CA trusted by the TOE.

For part ii: The evaluator configured a TLS client to modify a byte in the signature of the TLS client’s CertificateVerify record and verified the TOE rejected the connection when the validity of the CertificateVerify record from the client could not be established.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 5 [conditional]: The purpose of this test is to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g.

inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.

The TOE does not implement any administrative override for certificate validation failures.

FCS_TLSS_EXT.2.3
The evaluator shall send a client certificate with an identifier that does not match an expected identifier and verify that the server denies the connection.

The evaluator configured a TLS client to present a client certificate with an identifier that did not match the expected identifier. The evaluator verified that the TOE rejected the connection.

FCS_TLSS_EXT.2.4
Test 1a [conditional]: For TLS 1.3, the evaluator shall initiate a TLS session from a test TLS client and examine the Certificate Request message sent by the TSF. The evaluator shall verify the Certificate Request message contains the signature_algorithms extension and optionally the signature_algorithm_cert extension. If the signature_algorithms_cert extension exists, then the evaluator shall verify the SignatureScheme values within the signature_algorithms_cert extensions match the selections specified in the requirement. If only the signature_algorithms extension exists, then the evaluator shall verify the SignatureScheme values within the signature_algorithms extensions match the selections specified in the requirement. To view the Certificate Request message in TLS 1.3, the message will need to be decrypted.

This test is not applicable as the TOE does not utilize TLS 1.3 for the TLS server functionality.

FCS_TLSS_EXT.2.4
Test 2 [conditional]: For TLS 1.2, the evaluator shall initiate a TLS session from a test TLS client and examine the Certificate Request message sent by the TSF. The evaluator shall verify the Certificate Request message contains a list of supported_signature_algorithms. The evaluator shall verify the SignatureAndHashAlgorithm values within the supported_signature_algorithms list match the selections specified in the requirement.

The evaluator examined the CertificateRequest record sent by the TOE and verified that the list of Signature Algorithms specified in the CertificateRequest is consistent with the values specified in the ST.

FCS_TLSS_EXT.2.4
Test 3: The evaluator shall establish a TLS connection and perform client authentication with a certificate chain using each of the SignatureSchemes (TLS 1.3) or SignatureAndHashAlgorithm (TLS 1.2) specified by the requirement. The evaluator shall ensure the signature used in the certificate is consistent with the value being tested.

For each of the Signature Algorithms specified, the evaluator presented a leaf certificate that is signed with the Signature Algorithm. The evaluator verified that the TOE accepted the connection in each case and successfully completed the TLS connection.

2.3 Identification and Authentication (FIA)

2.3.1 FIA_AFL.1 Authentication Failure Management

2.3.1.1 TSS Activities

> 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.

Section 6.3 of [ST] states the TOE enforces a lockout mechanism that will trigger if an administrator-configured number between 1 and 10 of consecutive failed attempts is reached. The TOE keeps track of failed authentication attempts via internal counters. The lock can be configured to last a specified amount of time (0 – 60 minutes) during which providing the correct credentials will still not allow access (i.e. locked out). A setting of "0" will lock out the user indefinitely. In this case, the Administrator must unlock the locked user.

This section also states that this behavior applies to password-based authentication only because public key authentication cannot be brute forced in the same manner.

> 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).

Section 6.3 of [ST] states that in the evaluated configuration, it is required that at least one administrator, preferably the Superuser role (predefined 'admin' account), is configured with public-key authentication for SSH to prevent a denial of service due to locked accounts.

2.3.1.2 Guidance Activities

> 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.

Section 7.6 of [CCECG] provides instructions to configure the number of successive unsuccessful authentication attempts and time period, using the 'set deviceconfig setting management admin-lockout' command. The instructions note that setting the failure threshold to 0 means that lockout is not enforced, and setting duration to 0 means that the lockout persists indefinitely unless manually resolved. The guidance includes a warning not to set the failure threshold to 0 in the evaluated configuration and to use an SSH key for the default administrator account to prevent a denial of service of the management interface.

> 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.

Section 7.6 of [CCECG] explicitly requires the use of an SSH key as a fallback authentication mechanism for an administrator to prevent intentional or unintentional denial of service through the password lockout mechanism.

2.3.1.3 Test Activities

> 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):
>
> **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.

The evaluator configured the TOE to lock a user out after a series of unsuccessful authentication attempts for a period of time. The evaluator then presented incorrect authentication credentials to the TOE and verified that the TOE locked the user account once the configured number of unsuccessful authentication attempts was met. The evaluator verified that once the user was locked the user could no longer authenticate to the TOE with the valid credentials. The evaluator verified the TOE enforced the time period by attempting to authenticate to the TOE just before the time period expired and observed the TOE rejected the authentication attempt. The evaluator verified that the TOE granted the user access again once the lock out time period expired, and the user presented valid credentials.

> **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).
>
> 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.

The time-based component of the selection has been tested in FIA_AFL.1 Test 1.

2.3.2 FIA_PMG_EXT.1 Password Management

2.3.2.1 TSS Activities

The evaluator shall check that the TSS:

a. lists the supported special character(s) for the composition of administrator passwords.
b. to ensure that the minimum_password_length parameter is configurable by a Security Administrator.
c. lists the range of values supported for the minimum_password_length parameter. The listed range shall include the value of 15.

[ST] section 6.3 lists the supported special characters and states that the minimum password length can be configured to a value from 8-15 characters, with the maximum password being a fixed 31 characters.

Note in FIPS-CC mode, the minimum password length cannot be configured below 8. The maximum password length is 63 characters. For example, if the administrator configures the minimum password length as 15, they can only create passwords from length of 15 to 63.

2.3.2.2 Guidance Activities

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

Section 7.3.4 of [CCECG] states passwords can be composed of uppercase, lowercase, numbers, and special characters, and provides recommendations on the composition of strong passwords.

Section 7.7 of [CCECG] states how to set the minimum password length, and indicates that the minimum length can be set from anywhere between 8 and 15 characters.

2.3.2.3 Test Activities

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.

The evaluator composed a set of passwords that met the configured minimum password length and together covered all the characters claimed to be supported by the TOE. The evaluator verified TOE accepted each password that met the specified minimum requirements.

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.

The evaluator verified the TOE enforced the configured minimum password length. The evaluator found no character which could be input to the TOE that is not accepted by the TOE.

2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

2.3.3.1 TSS Activities

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

Section 6.3 of [ST] states administrators can logon to the TOE’s CLI using a secure connection from an SSH client. The TOE supports username for identification and password (defined internal to the TOE) or SSH public key for authentication. A logon successful note is provided when the correct credentials are used (username and password or SSH key) that matches a defined account on the TOE.

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

Section 6.3 of [ST] identifies that the only functionality the TOE will perform without authentication is to display the warning banner or respond to an ICMP request.

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

This evaluation activity is not applicable to the TOE because the TOE is not distributed.

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

This evaluation activity is not applicable to the TOE because the TOE is not distributed.

2.3.3.2 Guidance Activities

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.

[ST] defines the supported authentication mechanisms as password and SSH public key. For passwords, section 6 of [CCECG] states the administrator must change any default password. Section 6.3 of [CCECG] describes the process for doing this. To ensure subsequent passwords are of adequate strength, section 7.7 of [CCECG] describes how to change the minimum password length.

Section 7.3.2 of [CCECG] describes how to configure the supported authentication mechanisms for individual user accounts. If the administrator configures a user account to support SSH public key authentication, section 6.7 describes how to create an RSA key pair to use this function.

Section 7.5 of [CCECG] describes how to configure the text of the login banner the TOE displays prior to authentication. Section 5 of [CCECG] notes the only pre-authentication functions the TOE provides are the ability to view the login banner and to interact with the device using ICMP. No other un-authenticated functionality is configurable per [ST].

2.3.3.3 Test Activities

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:

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

The evaluator verified that the TOE validated the presented I&A information prior to granting access to the TOE and only grants a user access when the correct username and password/public key are presented and rejects access in all other cases.

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:

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.

The evaluator confirmed that the TOE would respond to ICMP request messages and present the warning banner configured in conjunction with FTA_TAB.1 testing prior to user authentication. The evaluator performed an nmap scan of the TOE and confirmed the only available services other than ICMP echo and display of an access banner are SSH and TLS, both of which require authentication and are documented as services provided by the TOE.

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.

This test is not applicable as the TOE claims no local authentication mechanisms.

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

This test is not applicable to the TOE because the TOE is not a distributed TOE.

2.3.4 FIA_X509_EXT.1/Rev X.509 Certificate Validation

2.3.4.1 TSS Activities

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).

Section 6.3 of [ST] states that X.509 certificates are used for TLS connections. This section also describes the rules for validating certificates (including how certificate path validation of three or more links is enforced). When importing a CA certificate to the TOE's trust store, it is checked for the validity of the certificate chain and the presence of the basicConstraints flag. When validating the import of a signed CSR response the TOE validates that the response chains to a CA certificate that is installed in the TOE’s trust store.

The TOE supports Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP) status verification for certificate profiles. In the evaluated configuration, the syslog connection implements OCSP for status verification for the certificate. The connection to the firewalls can use either CRLs or OCSP.

Section 6.3 of [ST] states that Certificate Revocation Lists (CRLs) are supported for certificate revocation checking. It also states that a certificate is checked for revocation status the first time it is used, and once validated, the status is cached for one hour. The TOE downloads and caches the last-issued CRL for every CA listed in the trusted CA list of the TOE. Caching only applies to validated certificates; if a TOE never validated a certificate, the TOE cache does not store the CRL for the issuing CA.

The TLS session for syslog is blocked when the certificate status is unknown or cannot be determined. This is the default behavior for syslog connections and cannot be changed. When configuring the TLS sessions for Firewalls, the administrator may configure the profile whether or not to block connections when certificate status is unknown or cannot be determined.

The TOE does not use X.509v3 certificates for trusted updates or executable code integrity verification. [ST] Section 6.5 states that Palo Alto Networks public key is to verify the digital signature on an update image.

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.

[ST] section 6.3 states that a certificate is checked for revocation status the first time it is used, and once validated, the status is cached for one hour. The TOE downloads and caches the last-issued CRL for every CA listed in the trusted CA list of the TOE. Caching only applies to validated certificates; if a TOE never validated a certificate, the TOE cache does not store the CRL for the issuing CA. Also, the cache only stores a CRL until it expires (one hour period). The TOE supports CRLs only in Distinguished Encoding Rules (DER) or PEM format.

No difference in the handling of certificate chains vs leaf certificates is identified.

2.3.4.2 Guidance Activities

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.

Section 6.8 of [CCECG] states the check of validity of certificates takes place during the TLS handshake.

Section 6.8.1 of [CCECG] states the TOE automatically checks revocation status based on CRL information located in the certificate for Syslog connections.

Section 6.8.2 of [CCECG] describes how to configure the TOE and environment so the TOE will check the revocation status of the firewall’s client certificate using CRLs or OCSP.

The TOE does not use X.509v3 certificates for trusted updates or executable code integrity verification, instead using the Palo Alto Networks public key to verify the digital signature on an update image (see Section 7.8 of [CCECG]) and using an HMAC-SHA-256 key and ECDSA public key to verify software integrity during power-up (see Section 7.9 of [CCECG]). As such, the TOE does not use or check for certificates with the Code Signing purpose specified in the extendedKeyUsage field.

2.3.4.3 Test Activities

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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). The evaluator shall perform the following tests for FIA_X509_EXT.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:

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).

The evaluator architected a TLS connection with the TOE such that the intermediate CA certificate which signs the leaf certificate is required to be provided with the leaf certificate. The evaluator provided both the leaf and intermediate CA certificate and verified the TOE accepted the connection.

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.

The evaluator provided just the leaf certificate and verified the TOE rejected the connection.

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

The evaluator verified that the TOE does not accept expired certificates when presented by a TLS server or a TLS client for mutual authentication and terminated the connection when the expired certificate was presented.

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 shall then attempt 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.

The evaluator verified the TOE validates the current revocation status of presented leaf and CA certificates when the certificate is present for TLS client or TLS server functionality. The evaluator verified the TOE accepts certificates which are valid at the time of the check and rejects leaf and CA certificates which are revoked at the time of the check. The evaluator verified the TOE uses OCSP for TLS client and TLS server functionality, and CRL for the TLS server functionality.

Test 4a: [conditional] If OCSP is selected, the evaluator shall configure an authorized responder or use a man-in-the-middle tool to use a delegated OCSP signing authority to respond to the TOE's OCSP request. The resulting positive OCSP response (certStatus: good (0)) shall be signed by an otherwise valid and trusted certificate with the extendedKeyUsage extension that does not contain the OCSPSigning (OID 1.3.6.1.5.5.7.3.9). The evaluator shall verify that the TSF does not successfully complete the revocation check.

Note: Per RFC 6960 Section 4.2.2.2, the OCSP signature authority is delegated when the CA who issued the certificate in question is NOT used to sign OCSP responses.

The evaluator verified that the TOE checks the validity of the provided OCSP response and rejects OCSP responses which are not signed by a certificate that is permitted to sign OCSP responses (e.g. the certificate signing the response lacks the OCSPSigning EKU). This was verified for both the TLS client and TLS server functionality.

Test 4b: [conditional] If CRL is selected, the evaluator shall present an otherwise valid CRL signed by a trusted certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

The evaluator verified the TOE checks the validity of the provided CRL and rejects CRL files which are signed by a certificate that is not permitted to sign CRL files (e.g. the certificate signing the CRL lacks the CRLSign key usage bit). This was verified for the TLS server functionality.

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.)

The evaluator confirmed that the TOE would not establish a TLS connection with a certificate modified in the first eight bytes when presented by a TLS server or a TLS client.

Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC 5280 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.)

The evaluator confirmed that the TOE would not establish a TLS connection with a certificate modified in the last byte when presented by a TLS server or a TLS client.

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.)

The evaluator confirmed that the TOE would not establish a TLS connection with a certificate modified in the public key when presented by a TLS server or a TLS client.

Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The following tests are run when a minimum certificate path length of three certificates is implemented:

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.

The evaluator installed the ECDSA root CA into the TOE’s trust store. The evaluator then configured a TLS peer to present an end entity certificate that was signed by an intermediate CA that uses namedcurve public key parameters in addition to the intermediate CA required to complete the chain. The evaluator observed that the TOE accepted the connection.

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.

The evaluator installed the ECDSA root CA into the TOE’s trust store. The evaluator then configured a TLS peer to present an end entity certificate that was signed by an intermediate CA that used explicit curve public key parameters in addition to the intermediate CA required to complete the chain. The evaluator observed that the TOE rejected the connection.

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.

The evaluator previously imported a root ECDSA certificate. The evaluator then attempted to import two different intermediate CA certificates signed by the root CA, one where the certificate contained the public key as a named curve value and one where the public key was defined as an explicit curve parameter value. The evaluator verified that the TOE only permitted the named curve certificate to be imported and rejected the explicit curve intermediate CA certificate.

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.

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).

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).

Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator shall confirm 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).

The evaluator presented a leaf certificate to the TOE which is signed by an intermediate CA certificate that lacks the basicConstraints extension. The evaluator verified the TOE rejected the connection when presented to the TLS client and TLS server interfaces of the TOE.

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 shall confirm 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).

The evaluator presented a leaf certificate to the TOE which is signed by an intermediate CA certificate that contains basicConstraints extension set to FALSE. The evaluator verified the TOE rejected the connection when presented to the TLS client and TLS server interfaces of the TOE.

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).

The evaluator verified the functionality of the TOE's handling of certificates over both the TLS server and TLS client interfaces of the TOE, as these are the only interfaces of the TOE that validate certificates this portion has been successfully completed.

2.3.5 FIA_X509_EXT.2 X.509 Certificate Authentication

[ST] iterates this SFR because the TOE has two separate X.509 certificate authentication implementations, each with their own security-relevant characteristics. Implementation differences between the two are noted below where applicable.

2.3.5.1 TSS Activities

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.

Section 6.3 of [ST] describes the TOE's usage of X.509 certificates for TLS authentication. It is implicit that the TOE has its own TLS client/TLS server certificate that it presents to remote entities, and the certificate it uses to validate the remote entity is the certificate that is provided to it during establishment of the trusted channel. Similarly, any intermediate/root CAs used by the TOE are implicit in the signer of any certificate that is presented to it.

To use only a specific trusted certificate, the Administrator must specify only that certificate in the Certificate Profile and tie that Profile to a TLS connection.

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.

Section 6.3 of [ST] states a TLS session for syslog is blocked when the certificate status is unknown or cannot be determined. This is the default behavior for syslog connections and cannot be changed. When configuring the TLS sessions for firewalls, the administrator may configure the profile whether or not to block connections when certificate status is unknown or cannot be determined.

2.3.5.2 Guidance Activities

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.

Section 6.8 of [CCECG] provides guidance to the administrator configuring secure connections with a syslog server (section 6.8.1 of [CCECG]) and firewall devices (section 6.8.2 of [CCECG]). The guidance describes the configuration required both on the TOE and in the operating environment to enable the TOE to use X.509v3 certificates to authenticate TLS connections, as both a TLS client authenticating an external TLS server (syslog server) and a TLS server authenticating an external TLS client (firewall device).

Section 6.8.1 of [CCECG] states, for connections to the syslog server, the TOE automatically drops the connection attempt if the revocation status of the syslog server certificate cannot be determined. This behavior is not configurable.

Section 6.8.2 of [CCECG] describes how to configure the TOE to either block or allow a connection with a firewall device if the revocation status of the firewall's certificate cannot be determined, using the 'block-unknown-cert' flag in the certificate profile.

2.3.5.3 Test Activities

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

a. Test 1: 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.

The evaluator performed this test for both the TLS client function and the TLS server function of the TOE.

For the TLS client, the evaluator configured the TOE to connect to a TLS peer that presented a certificate with revocation information present on it. The evaluator turned the revocation responder off. The evaluator observed that the TOE did not allow the connection to complete while the revocation responder was unavailable.

For the TLS server, the evaluator configured the TOE to block the connection if the TOE could not verify the certificate's revocation status during the connection attempt and ensured the revocation responder was still off. The evaluator caused a TLS client to attempt a connection to the TOE while presenting a certificate containing revocation status provider information and observed that the TOE rejected the connection. The evaluator then configured the TOE to accept the connection if the revocation provider could not be accessed and observed that the TOE accepted the connection.

2.3.6 FIA_X509_EXT.3 X.509 Certificate Requests

2.3.6.1 TSS Activities

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.

[ST] does not select "device-specific information" so this evaluation activity is N/A to the TOE.

2.3.6.2 Guidance Activities

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.

Section 6.8.1 of [CCECG] describes the use of the 'request certificate' command to generate a CSR. Consistent with the claims made by [ST], this command includes the 'country-code,' 'organization,' and 'name' parameters.

2.3.6.3 Test Activities

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.

The evaluator generated a certificate request on the TOE and verified it was in the correct format and contained all of the information specified by the Security Target.

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.

The evaluator attempted to import the signed response to the certificate request onto the TOE without the trusted CA imported and confirmed that the import was denied. The evaluator then imported the correct trusted CA and attempted to import the signed response and confirmed that the certificate was successfully imported.

2.4 Security Management (FMT)

General requirements for distributed TOEs

TSS

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

The TOE is not distributed so this evaluation activity is not applicable.

General requirements for distributed TOEs

Guidance Documentation

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

The TOE is not distributed so this evaluation activity is not applicable.

General requirements for distributed TOEs

Tests

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).

The TOE is not distributed so this evaluation activity is not applicable.

2.4.1 FMT_MOF.1/ManualUpdate Management of Functions Behavior

2.4.1.1 TSS Activities

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

The TOE is not distributed so this evaluation activity is not applicable.

2.4.1.2 Guidance Activities

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).

Section 7.8 of [CCECG] describes the steps necessary to perform a manual software update and notes that a reboot must occur after the update has been installed.

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).

The TOE is not distributed so this evaluation activity is not applicable.

2.4.1.3 Test Activities

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

The evaluator attempted to log into the TOE using a username of a command to execute an update of the TOE. The evaluator observed that the TOE treated the username as a proper username and did not attempt to execute the command.

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

The evaluator verified that the TOE permitted an authenticated Security Administrator to execute an update to the TOE. This test is performed in conjunction with FPT_TUD_EXT.1.

2.4.2 FMT_MTD.1/CoreData Management of TSF Data

2.4.2.1 TSS Activities

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

Section 6.3 of [ST] states that there are no security functions that are available to administrators prior to login aside from displaying the warning banner. The TOE also responds to ICMP in this state but that is a networking function and not an administrative one.

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

Section 6.4 of [ST] states that role-based privileges on the CLI are used to ensure that only authorized administrators can configure the TOE functions such as updating the TOE, managing X.509v3 certificates in the trust store, and manipulating TSF data.

2.4.2.2 Guidance Activities

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.

The following lists the security management functions for the TOE as claimed by [ST] and where in the vendor's documentation the usage of these functions is described:

- Ability to administer the TOE remotely - [CCECG] section 5.1.1 (for instructions on how to access the TOE locally and [CCECG] section 6.1 (for instructions on how to configure the whitelist so that administration is only permitted from approved locations).
- Ability to configure the access banner - [CCECG] section 7.5.
- Ability to configure the remote session inactivity time before session termination - [CCECG] section 7.6.
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates - [CCECG] section 7.8.
- Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1 - [CCECG] section 7.5.
- Ability to configure the cryptographic functionality – [CCECG] section 6.2 (enabling FIPS-CC mode configures this)
- Ability to configure thresholds for SSH rekeying – [CCECG] Section 6.5.
- Ability to set the time which is used for time-stamps - [CCECG] section 7.4.
- Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors - [CCECG] section 6.8.1 describes how to generate internal certificates, generate certificate signing requests, and import CA and leaf certificates. As noted in section 4 of [CCECG], importing CA certificates or generating CA certificates internally will implicitly set them as trust anchors.
- Ability to generate Certificate Signing Request (CSR) and process CA certificate response – [CCECG] Section 6.8.1.

- Ability to configure the authentication failure parameters for FIA_AFL.1 - [CCECG] section 7.6
- Ability to manage the trusted public keys database - [CCECG] section 6.6.

[CCECG] states that configuration information is only available to properly authenticated administrators.

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.

The TOE supports handling of X.509v3 certificates and provides a trust store. Per [ST], the TOE uses a KEK to protect stored certificate data and no additional configuration steps are required. Sections 6.8.2 and 7.2 of [CCECG] describe the process for loading certificates, including CA certificates. The ‘request certificate generate’ command described in section 6.8.1 of [CCECG] indicates that the ‘ca yes’ parameter is used to designate a CA certificate as a trust anchor.

2.4.2.3 Test Activities

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.

The evaluator verified all management functions were exercised under other SFRs using the CLI.

2.4.3 FMT_SMF.1 Specification of Management Functions

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.

2.4.3.1 TSS Activities (containing also requirements on Guidance Documentation and Tests)

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).

Section 6.4 of [ST] lists the supported management functions. It states the CLI provides the administrator the ability to administer the TOE with all management functions. The evaluation activities for the relevant SFRs demonstrate the proper implementation of these functions.

Section 2.4.2.2 of this AAR above lists the supported management functions and where in the vendor documentation their use is described. The evaluation team observed during testing that the TOE provides all the management functions specified in FMT_SMF.1.

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.

The TOE does not support local administration.

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.

The TOE is not distributed so this evaluation activity is not applicable.

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

This evaluation activity is not applicable. “Configure local audit” is not selected in FMT_SMF.1.

2.4.3.2 Guidance Activities

See Section 2.4.4.1

2.4.3.3 Test Activities

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

The evaluator performed testing for each management function in conjunction with the related SFR:

- Ability to administer the TOE remotely;

Tested implicitly as the TOE only implements remote administration.

- Ability to configure the access banner;

Tested in conjunction with FTA_TAB.1

- Ability to configure the remote session inactivity time before session termination;

Tested in conjunction with FTA_SSL.3

- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;

Tested in conjunction with FPT_TUD_EXT.1

- Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;

Tested in conjunction with FIA_UIA_EXT.1 and FTA_TAB.1

- Ability to configure the cryptographic functionality;

Tested in conjunction with FCS_TLSS_EXT.1.5

- Ability to configure thresholds for SSH rekeying;

Tested in conjunction with FCS_SSH_EXT.1.8

- Ability to set the time which is used for time-stamps;

Tested in conjunction with FPT_STM_EXT.1

- Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors;

Tested in conjunction with FIA_X509_EXT.1

- Ability to generate Certificate Signing Request (CSR) and process CA certificate response;

Tested in conjunction with FIA_X509_EXT.3

- Ability to configure the authentication failure parameters for FIA_AFL.1;

Tested in conjunction with FIA_AFL.1

- Ability to manage the trusted public keys database;

Tested in conjunction with FIA_UIA_EXT.1

2.4.4 FMT_SMR.2 Restrictions on Security Roles

2.4.4.1 TSS Activities

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

[ST] section 6.4 identifies the pre-defined superuser supported by the TOE. The administrator role (Superuser) is considered the Security Administrator as defined in the [NDcPP] for the purposes of the ST. All roles can administer the TOE securely, and a user account can only be assigned one role at a time.

2.4.4.2 Guidance Activities

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.

Sections 6.2, 6.4, 6.5, and 6.6 of [CCECG] describe how to configure the TOE to access the management interface over the remote SSH trusted path in the manner specified by [ST]. Once configured, section 5.1.1 of [CCECG] describes how to administer the TOE remotely.

2.4.4.3 Test Activities

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

All configuration activities and management are done through a remote SSH session and issuing CLI commands to the TOE. As the TOE only supports one interface for administration, this test is implicitly met.

2.5 Protection of the TSF (FPT)

2.5.1 FPT_APW_EXT.1 Protection of Administrator Passwords

2.5.1.1 TSS Activities

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.

Section 6.5 of [ST] states that password data is obfuscated through the lack of a dedicated interface to view stored password data and through the SHA-256 hashing of passwords. It also states that certificates (used for authentication) and their associated key data is stored in a PKCS#12 file which stores the x.509 certificate and encrypted private key.

2.5.1.2 Guidance Activities

None

2.5.1.3 Test Activities

None

2.5.2 FPT_SKP_EXT.1 Protection of TSF Data (for Reading of All Pre-shared, Symmetric, and Private Keys)

2.5.2.1 TSS Activities

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

Section 6.5 of [ST] states that all secret and private key data is stored using 256-bit AES encryption by a Master Key, and that there is no interface by which a user can view the Master Key.

2.5.2.2 Guidance Activities

None

2.5.2.3 Test Activities

None

2.5.3 FPT_STM_EXT.1 Reliable Time Stamps

2.5.3.1 TSS Activities

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.

Section 6.5 of [ST] states that the clock is used for audit record time stamps, measuring session activity for termination, certificate validity checking, timing administrator lockout due to excessive failed authentication attempts, and for cryptographic operations based on time/date.

The time is maintained and considered reliable in the context of each of the time related functions. The TOE is a hardware appliance that includes a hardware-based real-time clock. The TOE’s embedded OS manages the clock and exposes administrator clock-related functions such as set time.

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.

The evaluation activity is not applicable. The TOE is a hardware appliance and not a virtual system.

2.5.3.2 Guidance Activities

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

Section 7.4.1 of [CCECG] instructs the administrator how to set the time manually using the ‘set clock’ CLI command. The TSF does not claim support for an NTP server.

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.

The TOE does not include a vND and does not rely on an underlying VS as a time source. Therefore, this activity is not applicable to the TOE.

2.5.3.3 Test Activities

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

The evaluator queried the current time on the TOE, then attempted to change the time. The evaluator queried the time again and verified that the time successfully changed to the value specified by the evaluator.

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

This test is not applicable because the TOE does not support use of an NTP server.

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.

This test is not applicable because the TOE is not a virtualized device, thus there is no underlying VS.

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.

This test is not applicable because the TOE consists of a single standalone device and therefore only has time source.

2.5.4 FPT_TST_EXT.1 TSF Testing

2.5.4.1 TSS Activities

Modified by TD0836

The evaluator shall examine the TSS to ensure that it details **each of** the self-tests that are **identified by the SFR** ~~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. If more than one failure response is listed in FPT_TST_EXT.1.2, the evaluator shall examine the TSS to ensure it clarifies which response is associated with which type of failure.

Section 6.5 of [ST] lists the self-tests performed by the TOE. All self-tests are either cryptographic known-answer tests or conditional tests, either for cryptography or the firmware of the cryptographic module. [ST] argues that this is sufficient to ensure correct functionality of the TSF because the self-tests encompass the cryptographic functionality and the integrity of the entire TOE software executable code.

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

The TOE is not distributed so this evaluation activity is not applicable.

2.5.4.2 Guidance Activities

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.

Section 7.9 of [CCECG] states if any TSF self-test fails, the TOE enters an error state and does not operate until the error is resolved. The administrator can attempt to resolve the issue by rebooting the TOE. The guidance advises the administrator, in the event of continued failures, to contact Palo Alto Networks Support and provides an email address and phone number.

The description in the guidance of the possible errors that may result from the self-tests corresponds to the errors described in the TSS.

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.

The TOE is not distributed so this evaluation activity is not applicable.

2.5.4.3 Test Activities

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.

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

a. [FIPS 140-2], Section 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], Section 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.

The evaluator verified the TOE reported that self-tests were executed successfully for the cryptographic functions utilized by the TOE and the integrity of the TOE.

Modified by TD0836

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~~ **according to the SFR and in agreement with the descriptions in the TSS**.

The evaluator verified the TOE executed self-tests during the start-up sequence.

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.

The TOE is not distributed so this portion of the evaluation activity is not applicable.

2.5.5 FPT_TUD_EXT.1 Trusted Update

2.5.5.1 TSS Activities

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

Section 6.5 of [ST] identifies the CLI command that is used to show the current software version.

The TSF does not contain a delayed activation mechanism for downloaded updates, so this is not discussed in the TSS.

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

Section 6.5 of [ST] identifies the commands used to check for updates and then download them. As part of the download activity, the update's 2048-bit RSA digital signature is checked. The update is only installed if the signature verification is successful. The TOE does not use a published hash for update integrity verification.

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.

[ST] does not claim automatic checking or application of updates so this activity is N/A.

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

The TOE is not distributed so this evaluation activity is not applicable.

2.5.5.2 Guidance Activities

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

Section 7.8 of [CCECG] describes how to query the currently active version of software using the ‘show system info’ CLI command. The TOE does not provide a capability to install a trusted update with a delayed activation.

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

Section 7.8 of [CCECG] states the TOE automatically validates the update’s digital signature during installation of the update, and the install process will automatically terminate if the signature is invalid.

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.

The TOE is not distributed so this evaluation activity is N/A.

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.

The TOE is not distributed so this evaluation activity is N/A.

If this was information was not provided in the TSS: If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the Guidance Documentation contains a description of how the certificates are contained on the device. The evaluator shall also ensure that the Guidance Documentation describes how the certificates are installed/updated/selected, if necessary.

[ST] does not claim the use of a certificate-based mechanism for software updates so this evaluation activity is N/A.

2.5.5.3 Test Activities

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

The evaluator observed the TOE's current version, executed an update of the TOE, and then observed the current version again. The TOE was verified to have been updated to the new version.

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

i. A modified version (e.g. using a hex editor) of a legitimately signed update
ii. An image that has not been signed
iii. An image signed with an invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate signature)
iv. The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify that both the current

version and most recently installed version, reflect the same version information as prior to the update attempt.

The evaluator attempted to update the TOE with software images that were modified, unsigned, and had an invalid/modified signature, and confirmed that the TOE did not successfully update.

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

The TOE only supports one method of update; thus this portion of the activity has been implicitly met.

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

This portion is not applicable as the TOE is not a distributed TOE.

2.6 TOE Access (FTA)

2.6.1 FTA_SSL.3 TSF-initiated Termination

2.6.1.1 TSS Activities

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

[ST] section 6.6 describes the session termination behavior. The inactivity time period is a configurable value between 1 and 1,440 minutes, with a default of 60 minutes.

2.6.1.2 Guidance Activities

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

Section 7.6 of [CCECG] states the administrator can configure an idle session timeout for CLI users that access the TOE remotely via SSH. It includes an example of the use of the ‘set deviceconfig setting management idle-timeout’ command to configure the inactivity timeout period (in minutes).

2.6.1.3 Test Activities

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

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

The evaluator configured the TOE to have various inactivity timeout values. The evaluator then authenticated to the TOE and left the session inactive. The evaluator verified when the configured inactivity period elapsed the user was logged out and had to re-authenticate to the TOE.

2.6.2 FTA_SSL.4 User-initiated Termination

2.6.2.1 TSS Activities

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

[ST] section 6.6 states that the TOE provides the function to logout (or terminate) user sessions as directed by the user.

2.6.2.2 Guidance Activities

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

Section 5.1.2 of [CCECG] states the 'exit' command is used to terminate an interactive CLI session.

2.6.2.3 Test Activities

Test 1 [conditional]: If the TOE supports local administration, the evaluator shall initiate an interactive local session with the TOE. The evaluator shall then follow the guidance documentation to exit or log off the session and observes that the session has been terminated.

This test is not applicable as the TOE does not support local administration of the TOE.

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

The evaluator logged in, over the remote administration channel, then used the "exit" command to log out of the TOE. The evaluator confirmed the user was logged out and needed to re-authenticate to regain access to the TOE.

2.6.3 FTA_TAB.1 Default TOE Access Banners

2.6.3.1 TSS Activities

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

Section 6.6 of [ST] states that the TOE displays a configurable warning banner on the CLI prior to administrator authentication.

2.6.3.2 Guidance Activities

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

Section 7.5 of [CCECG] describes how to configure the banner text using the 'set deviceconfig system login-banner' command.

2.6.3.3 Test Activities

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

The evaluator configured the TOE access banner and verified the configured banner was displayed prior to authentication.

2.7 Trusted Path/Channels (FTP)

2.7.1 FTP_ITC.1 Inter-TSF Trusted Channel

2.7.1.1 TSS Activities

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.

Section 6.7 of [ST] identifies the trusted channels used by the TOE. Specifically, the TOE supports TLS for syslog server connections and connections with Palo Alto firewalls. For each use of TLS, this section indicates whether the TOE acts as a client or a server for the connection.

2.7.1.2 Guidance Activities

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.

Section 6.8 (and subsections) of [CCECG] describes how to configure the TLS trusted channels that are described in [ST]. This also includes instructions in the event of an unintentional termination.

2.7.1.3 Test Activities

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.

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.

The only protocol utilized by the TOE is TLS which has been tested under FCS_TLSS_EXT.1 and FC_TLSC_EXT.1.

The evaluator verified that communication with each entity specified was tested as part of the evaluation.

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.

The TOE only initiates connections to the remote syslog server. The evaluator verified that the TOE successfully initiated the connection to the remote syslog server.

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

The TOE only utilizes the TLS protocol and does not permit the TLS_NULL_WITH_NULL_NULL ciphersuite to be utilized for establishing a connection. Thus, the evaluator has verified that the channel data is not sent in plaintext.

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

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

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

In the case where the TOE is able to detect TOE external interruption (such as a cable being physically removed or a virtual connection being disabled), another network device shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall be external to the TOE (i.e., by manipulating the test environment and not by TOE configuration change).

The evaluator interrupted the connection between the TOE and the remote syslog server at a core switch for a short period of time, ~10 seconds, and then reconnected the connection. The evaluator observed that after the connection was re-established the traffic from the TOE to the remote syslog server was sent in a protected and encrypted manner.

Next, the evaluator interrupted the connection between the TOE and the remote syslog server at a core switch for a longer period of time, ~35 minutes, and then reconnected the connection. The evaluator observed that after the connection was re-established the traffic from the TOE to the remote syslog server was sent in a protected and encrypted manner.

Further assurance activities are associated with the specific protocols.

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.

The TOE is not distributed so this evaluation activity is not applicable.

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.

The evaluator used the provided application layer timeout for the evaluation.

2.7.2 FTP_TRP.1/Admin Trusted Path

2.7.2.1 TSS Activities

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.

Section 6.7 of [ST] states that the TOE supports an SSH trusted path that is used for remote authentication. 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.

2.7.2.2 Guidance Activities

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

Section 5.1.1 of [CCECG] identifies the remote administrative protocol as SSH and that this protocol is enabled by default. Sections 6.2, 6.4, 6.5, and 6.6 of [CCECG] describe how to ensure that SSH is configured in a manner that conforms to [ST].

2.7.2.3 Test Activities

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.

This test was performed throughout testing by establishing an SSH session to the TOE and administering the TOE within that session as SSH is the only claimed protocol for administration.

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

The evaluator established an SSH session with the TOE and confirmed the channel data was encrypted as required by the SSH protocol.

Further assurance activities are associated with the specific protocols.

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.

The TOE is not distributed so this evaluation activity is not applicable.

3 Security Assurance Requirements

3.1 Class ASE: Security Target Evaluation

General ASE

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

3.1.1 ASE_TSS.1 TOE Summary Specification for Distributed TOEs

This section is N/A for this evaluation because the TOE is not distributed.

3.2 Class ADV: Development

3.2.1 ADV_FSP.1 Basic Functional Specification

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

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

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

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

3.2.1.1 ADV_FSP.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.

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

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

Section 2.2.1 of [ST] identifies the security relevant TSFIs as remote syslog server, external Palo Alto firewall device, and workstation (SSHv2 client). The TSS describes these logical interfaces as TLS trusted channels and SSH trusted path and defines their operation in terms of the relevant FCS and FTP requirements. Section 2.2.1 of [ST] also defines the external physical interfaces of the TOE in sufficient detail to determine their security relevance (e.g. noting that the DB-9 serial port and reserved USB ports are disabled in FIPS-CC mode identifies this as non-TSFI, and the power supply interface is obviously non-TSFI based on its intended use).

3.2.1.2 ADV_FSP.1 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.

The vendor developed [CCECG] specifically to address the security functionality as identified in [ST]. The Guidance activities for the individual SFRs demonstrate that the guidance documentation includes sufficiently detailed instructions to configure and use the TSFIs in the manner required by [ST]. This is demonstrated by the evaluation activities for FCS_SSHS_EXT.1, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, and FMT_MTD.1/CoreData.

3.2.1.3 ADV_FSP.1 Evaluation Activity

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

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

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

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

Section 6.7 of [ST] associates the TOE's remote logical interfaces with the FTP_ITC.1 and FTP_TRP.1/Admin SFRs. Additionally, this section identifies the trusted channel protocol and which end of the connection the TOE is (client or server) as needed to specifically associate each logical interface further with FCS_SSHS_EXT.1, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, or FCS_TLSS_EXT.2 as appropriate. The TOE also contains a local management interface, which is understood not to relate to FCS_SSHS_EXT.1.

Additionally, the intended usage of each logical interface can be inferred from the TSS to the extent that their applicability to other SFRs can be determined. Specifically, the syslog interface is also used in support of FAU_STG_EXT.1 and the management interface is used to enforce the various FMT requirements and supports the enforcement of the various FIA and FTA requirements through its usage.

3.3 Class AGD: Guidance Documents

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

Note that additional Evaluation Activities for the guidance documentation in the case of a distributed TOE are defined in Appendix B.4.2.1.

3.3.1 AGD_OPE.1 Operational User Guidance

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

In addition, the evaluator performs the EAs specified below.

3.3.1.1 AGD_OPE.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.

The [CCECG] is published with the Security Target at the https://www.niap-ccevs.org/ website. Section 1.3 of [CCECG] ("Documentation References") lists other documentation (comprising [ADMIN] and [WF-500-B HW REF]) referenced from [CCECG] that the vendor publishes on its web site. This includes URLs for each document. The evaluator verified the URLs link to the correct documents on the vendor web site. The distribution of the documentation provides a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration. As per NIAP policy, [CCECG] and the other supporting documents are posted on the NIAP Product Compliant List.

3.3.1.2 AGD_OPE.1 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.

According to [ST], the TOE solely refers to the Palo Alto Networks WF-500-B Appliances running version 11.1. [CCECG] (in section 1.2) and [Admin] refers explicitly to the WF-500-B appliance; specifically [WF-500-B HW REF].

[ST] section 2.2.1 lists the supported Operational Environment components (syslog server, external firewall, SSH client). [CCECG] describes support for these components and configuration of their interactions with the TOE.

3.3.1.3 AGD_OPE.1 Evaluation Activity

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

Section 6.2 of [CCECG] describes how to enable FIPS-CC mode and explicitly states this is required by the TOE.

3.3.1.4 AGD_OPE.1 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.

Section 1.1 of [CCECG] introduces the guidance documentation by warning the reader that only the security functionality claimed by the ST has been addressed by the evaluated configuration.

3.3.1.5 AGD_OPE.1 Evaluation Activity

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

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

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

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

Section 3.3.1.3 above addresses part a).

For part b), section 7.8 of [CCECG] describes the update process. Specifically, administrators use the TOE to check for updates made available on the Palo Alto support site. The 'request system software download' command is used to acquire an update. Once downloaded, the 'request system software install' command initiates the update process, which automatically checks the validity of the digital signature. This section notes the TOE's behavior in the event of an update failure.

Section 3.3.1.4 above addresses part c).

3.3.2 AGD_PRE.1 Preparative Procedures

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

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

In addition, the evaluator performs the EAs specified below.

3.3.2.1 AGD_PRE.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).

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

The evaluators reviewed the [WF-500-B HW REF] document and determined that it is written at a general level that is easily understood by a technical audience. This guide relates to the physical setup of the TOE. The evaluators reviewed [Admin] and [CCECG] and determined that they are also written at a level that is appropriate for the intended audience. These guides relate to the logical setup and operational administration of the TOE.

3.3.2.2 AGD_PRE.1 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.

The evaluators observed that [CCECG], [Admin], and [WF-500-B HW REF] identify the same TOE model that is claimed in [ST]. While conducting the review of the [CCECG] against the claimed SFRs, the evaluators observed that all environmental components were discussed.

3.3.2.3 AGD_PRE.1 Evaluation Activity

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

The evaluators reviewed [WF-500-B HW REF] and observed that it contains instructions for the physical deployment of the TOE hardware. The evaluators also reviewed [Admin] and [CCECG] and determined that they describe how set up the TOE's logical interfaces for initial use and to configure the external interfaces specified as the TOE's Operational Environment by [ST].

3.3.2.4 AGD_PRE.1 Evaluation Activity

> The evaluator shall examine the preparative procedures to ensure they include instructions on how to manage the TSF as a product and as a component of the larger Operational Environment in a manner that allows to preserve integrity of the TSF.
>
> The intent of this requirement is to ensure there exists adequate preparative procedures (guidance in most cases) to put the TSF in a secure state (i.e., evaluated configuration). AGD_PRE.1 lists general requirements, the specific assurance activities implementing it are performed as part of FMT_SMF.1, FMT_MTD.1 and FMT_MOF.1 series of SFRs

The evaluators observed that the TOE's documentation includes guidance on configuring the TOE's interactions with its operational environment where needed. This includes setting up trusted channels to the TOE's Operational Environment even though the data carried over those channels is outside the scope of [NDcPP]. This ensures that the TOE can be deployed properly in its intended context while being configured in a secure manner as claimed by [ST].

3.3.2.5 AGD_PRE.1 Evaluation Activity

> In addition the evaluator shall ensure that the following requirements are also met.
>
> The preparative procedures must
>
> a. include instructions to provide a protected administrative capability; and
>
> b. identify TOE passwords that have default values associated with them and instructions shall be provided for how these can be changed.

The evaluators reviewed [CCECG] and observed that the CLI uses SSH by default. However, [CCECG] also includes instructions for enabling CC-FIPS mode, which ensures that SSH is configured in the manner claimed by [ST]. It also includes instructions for additional configuration steps to ensure that the SSH configuration is consistent with [ST].

The evaluators reviewed [CCECG] and observed that section 6.3 states that the admin account's default password is 'paloalto' along with instructions for how to change this.

3.4 Class ALC: Life-Cycle Support

3.4.1 ALC_CMC.1 Labeling of the TOE

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

3.4.2 ALC_CMS.1 TOE CM Coverage

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

3.4.3 Systematic Flaw Remediation (ALC_FLR.3)

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

The evaluation team verified this through the completion of the ALC_FLR.3 work units described in the CEM. The results of this analysis were included in the proprietary ETR produced by the laboratory.

3.5 Class ATE: Tests

3.5.1 ATE_IND.1 Independent Testing – Conformance

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

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

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

Note that additional Evaluation Activities relating to evaluator testing in the case of a distributed TOE are defined in section B.4.3.1.

The evaluators developed a test plan ([Test]) to list all of the individual test evaluation activities for the TOE based on the claimed SFRs. The test plan lists, for each evaluation activity, the test results (including supporting evidence where necessary) and the testing verdict. In all cases, the tests were observed to be passing.

The TOE consists of a single Palo Alto WildFire WF-500-B appliance. All tests were executed on this appliance, thus no equivalency argument needs to be made.

Testing of the TOE was performed at the Leidos Accredited Testing and Evaluation Lab located in Columbia, Maryland between August 2024 and September 2025.

3.6 Class AVA: Vulnerability Assessment

3.6.1 AVA_VAN.1 Vulnerability Survey

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

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

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

3.6.1.1 AVA_VAN.1 Evaluation Activity (Documentation)

In addition to the activities specified by the CEM in accordance with Table 2, the evaluator shall perform the following activities.

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.

The developer shall provide documentation identifying the list of software and hardware components[7] that compose the TOE. Hardware components should identify compute-capable hardware components, at a minimum that must include the processor, and where applicable, discrete crypto ASICs, TPMs, etc. used by the TOE. Software components include applications, the operating system and other major components that are independently identifiable and reusable (outside of the TOE), for example a web server, protocol or cryptographic implementations, (independently identifiable and reusable components are not limited to the list provided in the example). This additional documentation is merely a list of the name and version number of the components and will be used by the evaluators in formulating vulnerability hypotheses during their analysis.

As per [ST] and the evidence used for the evaluation of AGD_OPE.1 and AGD_PRE.1, the evaluators identified the following materials:

- Processors used by the TOE: identified in Table 1 of [ST].
- Software components used by the TOE: identified in section 2.2 of [ST].
- Materials related to distributed TOE requirements are N/A because the TOE is not distributed.

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

a. documentation describing the allocation of requirements between distributed TOE components as in [NDcPP, 3.4]
b. a mapping of the auditable events recorded by each distributed TOE component as in [NDcPP, Table 2]
c. additional information in the Preparative Procedures as identified in the refinement of AGD_PRE.1 in additional information in the Preparative Procedures as identified in 3.4.1.2 and 3.5.1.2.

The TOE is not distributed.

3.6.1.2 AVA_VAN.1 Evaluation Activity

The evaluator shall formulate hypotheses in accordance with process defined in Appendix A. The evaluator shall document 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.

The evaluators conducted vulnerability research and penetration testing to determine the vulnerability of the TSF to attackers with Basic Attack Potential.

The evaluators conducted searches in public vulnerability repositories for the following Type 1 flaws based on the guidance specified in [ND-SD]:

- The list of software and hardware components that comprise the TOE
- The TOE name (including model information as appropriate).

The evaluation team performed a search of the following public vulnerability database:

- National Vulnerability Database (https://web.nvd.nist.gov/vuln/search)
- US-Cert (http://www.kb.cert.org/vuls/)
- Tipping Point Zero Day Initiative (https://www.zerodayinitiative.com/advisories/published/)
- Palo Alto Networks Security Advisories (https://security.paloaltonetworks.com/)

Specifically, the following search terms were used in these searches:

- The list of software and hardware components that comprise the TOE:
    - Processor:

- Intel Xeon Silver 4316
- Ice Lake microarchitecture
- Software:
  - WildFire 11.1
  - PAN-OS 11.1
  - Palo Alto Networks Crypto Module 11
  - Linux 4.18.0
  - OpenSSL 1.1.1g
  - OpenSSH 8.0p1.
- “Palo Alto Wildfire”, “Palo Alto Networks Wildfire”, and “WF-500-B” as variations of the TOE name.

The evaluators performed these searches most recently October 21, 2025.

Additionally, the evaluators performed fuzz testing of the TOE as specified in Section A.1.4 of [ND-SD]. The evaluators observed the TOE did not react adversely to the packets directed at the TOE or respond to the packets. This testing did not discover any vulnerabilities in the TOE.

The evaluation team determined that no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential. This information is documented in [VA].