Assurance Activities Report

for

Palo Alto Networks WF-500 WildFire 10.1

Version 1.0

2022-08-03

Prepared by:

Leidos Inc.
https://www.leidos.com/CC-FIPS140
Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, MD 21046

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

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
Kofi Owusu
Pascal Patin
Allen Sant

Contents

1 Introduction

This document presents results from performing evaluation activities associated with the evaluation of Palo Alto Networks WF-500 WildFire 10.1. 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 2.2, December 2019 [ND-SD].

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 CAVP 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 [ND-SD]. Rationale is included for those Technical Decisions that do not apply to this evaluation.

TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH

This TD is not applicable to the TOE. It modifies FCS_SSHC_EXT.1 but the TOE does not claim that SFR.

TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters

This TD is applicable to the TOE.

TD0634: NIT Technical Decision for Clarification required for testing IPv6

This TD is applicable to the TOE.

TD0633: NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance

This TD is not applicable to the TOE. It modifies FCS_IPSEC_EXT.1 but the TOE does not claim that SFR.

TD0632: NIT Technical Decision for Consistency with Time Data for vNDs

This TD is not applicable to the TOE. The TD adds a selection to FPT_STM_EXT.1.2 but the TSF does not claim the added functionality.

TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server

This TD is applicable to the TOE.

TD0592: NIT Technical Decision for Local Storage of Audit Records

This TD is not applicable to the TOE. It modifies introductory text related to optional audit-related SFRs the TOE does not claim.

TD0591: NIT Technical Decision for Virtual TOEs and hypervisors

This TD is not applicable to the TOE. It modifies a portion of an assumption that relates to virtual network devices. The TOE has no virtual implementations so that portion of the assumption does not apply to it.

TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3

This TD is applicable to the TOE. The TD modifies a selection option in FCS_CKM.2 that the ST claims.

TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e

This TD is applicable to the TOE. The TD adds a selection option in FCS_CKM.2 that the ST claims.

TD0572: NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers

This TD is a clarification of the intent of the FCS_TLSC_EXT and FTP_ITC.1 requirements and therefore is generally applicable to the TOE but does not change any of the requirements or evaluation materials.

TD0571: NiT Technical Decision for Guidance on how to handle FIA_AFL.1

This TD is a clarification of the intent of the requirement and therefore is generally applicable to the TOE but does not change any of the requirements or evaluation materials.

TD0570: NiT Technical Decision for Clarification about FIA_AFL.1

This TD is a clarification of the intent of the requirement and therefore is generally applicable to the TOE but does not change any of the requirements or evaluation materials.

TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7

This TD applies to the TOE and has been incorporated into the AAR.

TD0564: NiT Technical Decision for Vulnerability Analysis Search Criteria

This TD applies to the TOE and has been incorporated into the AAR.

TD0563: NiT Technical Decision for Clarification of audit date information

This TD is applicable to the TOE. The TD clarifies the interpretation of FAU_GEN.1 and as such only modifies an application note.

TD0556: NIT Technical Decision for RFC 5077 question

This TD applies to the TOE and has been incorporated into the AAR.

TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test

This TD applies to the TOE. The TD was resolved through the issuance of TD0556.

TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN

This TD applies to the TOE and has been incorporated into the AAR.

TD0546: NIT Technical Decision for DTLS - clarification of Application Note 63

This TD is not applicable to the TOE; the TD relates to DTLS functionality, which the TOE does not claim.

TD0538: NIT Technical Decision for Outdated link to allowed-with list

This TD is a semantic issue with the claimed PP that was corrected. It does not affect the ST or the evaluation of the TOE.

TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3

This TD applies to the TOE but does not affect any evaluation materials as it only clarifies the interpretation of an SFR in an application note.

TD0536: NIT Technical Decision for Update Verification Inconsistency

This TD applies to the TOE and has been incorporated into the AAR.

TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4

This TD does not apply to this TOE. The TD relates to NTP functionality, which the TOE does not claim.

TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1)

This TD applies to the TOE and has been incorporated into the AAR.

1.2 Evidence

[ST] Palo Alto Networks WF-500 WildFire 10.1 Security Target, Version 1.0, August 1, 2022

[CCECG] Palo Alto Networks Common Criteria Evaluated Configuration Guide (CCECG) for WildFire 10.1, Version 1.0, August 1, 2022

[Admin] WildFire Administrator's Guide Version 10.1, 21 May 2021

[HRG] WF-500 WildFire Appliance Hardware Reference Guide, February 29, 2016

[Test] Palo Alto Networks WF-500 WildFire 10.1 Common Criteria Test Report and Procedures For Network Device collaborative PP Version 2.2e, Version 1.1, August 1, 2022

[VA] Palo Alto Networks WF-500 WildFire 10.1 Vulnerability Assessment, Version 1.1, 3 August 2022

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.

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_TSS.1	Pass
ADV_FSP.1	Pass
AGD_OPE.1	Pass
AGD_PRE.1	Pass
ALC_CMC.1	Pass
ALC_CMS.1	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 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 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 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 all of the auditable events for NDcPP. In each case, a sample audit record is included. The table provides multiple examples of audit events when multiple failure conditions may occur to show that sufficient information is provided to the administrator to diagnose the cause of the failure (e.g., FIA_X509_EXT.1/Rev has “Client cert expired or revoked for peer,” “Certificate unknown for peer,” and “Certificate status is unavailable” messages).

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.

The evaluators reviewed [CCECG] and determined the table of auditable events in Chapter 4 contains sample audit records for management functions as described by FMT_SMF.1. Under FAU_GEN.1, it also identifies a sample audit record for "resetting passwords". Because FAU_GEN.1 and FMT_SMF.1 define all of the security-relevant manipulation of TSF data that is within scope of the [NDcPP], the association of these functions with auditable events is sufficient to determine that auditing is described in sufficient detail.

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 I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism.

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

The evaluator performed 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 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 the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access.

Section 6.1 of [ST] states when a log reaches the maximum size, new audit data overwrites the oldest 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 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.

[ST] section 6.1 indicates that the TOE is a standalone device that stores 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.

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

The evaluator shall examine the TSS to ensure that it details the behaviour of the TOE when the storage space for audit data is full. When the option ‘overwrite previous audit record’ is selected this description should include an outline of the rule for overwriting audit data. If ‘other actions’ are chosen such as

sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.

[ST] selects "overwrite previous audit records according to the following rule:" and completes the "rule for overwriting previous audit records" assignment in the SFR. Section 6.1 of [ST] states that when the log file reaches its maximum size, new audit data overwrites the oldest audit data.

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

Section 6.1 of [ST] states that audit data is transmitted to the remote audit server in real-time.

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

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 6.7.1 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 6.7.1 of [CCECG] also describes requirements for configuring the external syslog server. It directs the administrator to install or use syslog-ng 3.7 or later and describes how to configure syslog-ng and the external syslog server so the TOE can successfully export its audit records to the external syslog server.

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

Section 6.7.1 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 also ensure that the guidance documentation describes all possible configuration options for FAU_STG_EXT.1.3 and the resulting behaviour of the TOE for each possible configuration. The description of possible configuration options and resulting behaviour shall correspond to those described in the TSS.

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 data will overwrite the oldest audit data.

2.1.3.3 Test Activities

Testing of the trusted channel mechanism for audit will be performed as specified in the associated evaluation activities for the particular trusted channel mechanism. 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.

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

1) The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option 'drop new audit data' in FAU_STG_EXT.1.3).
2) The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option 'overwrite previous audit records' in FAU_STG_EXT.1.3)
3) The TOE behaves as specified (for the option 'other action' in FAU_STG_EXT.1.3).

The evaluator performed actions to generate audit events until the local audit trail was full and confirmed that the oldest events were overwritten when new events were generated.

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

This test is not applicable since FAU_STG_EXT.2/LocSpace is not claimed.

Test 4: For distributed TOEs, Test 1 defined above should be applicable to all TOE components that forward audit data to an external audit server. For the local storage according to FAU_STG_EXT.1.2 and FAU_STG_EXT.1.3 the Test 2 specified above shall be applied to all TOE components that store audit data locally. For all TOE components that store audit data locally and comply with FAU_STG_EXT.2/LocSpace Test 3 specified above shall be applied.
The evaluator shall verify that the transfer of audit data to an external audit server is implemented.

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

2.2 Cryptographic Support (FCS)

Section 6.2 of [ST] identifies the TOE's CAVP 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)		
FFC key pair generation (key size 2048 bits)	FIPS PUB 186-4	#A2137 (includes DSA KeyGen (FIPS186-4), ECDSA KeyGen (FIPS186-4), RSA KeyGen (FIPS186-4))
ECC key pair generation (NIST curves P-256, P-384, P-521)	FIPS PUB 186-4	
RSA key generation (key size 2048, 3072, 4096 bits)	FIPS PUB 186-4	
FFC Schemes using Diffie-Hellman Group 14	RFC 3526, Section 3	N/A (Vendor Assertion)
Cryptographic Key Establishment (FCS_CKM.2)		
RSA based key establishment	RSAES-PKCS1-v1_5	N/A (Vendor Assertion)
ECDSA based key establishment	NIST SP 800-56A	#A2137 (includes KAS-ECC-SSC Sp800-56Ar3 and KAS-FFC-SSC Sp800-56Ar3 component key exchange)
FFC based key establishment	NIST SP 800-56A	
Key establishment scheme using Diffie-Hellman Group 14	RFC 3526, Section 3	N/A (Vendor Assertion)
AES Data Encryption/Decryption (FCS_COP.1/DataEncryption)		
AES CBC, CTR, GCM (128, 256 bits)	AES as specified in ISO 18033-3 CBC, CTR as specified in ISO 10116 GCM as specified in ISO 19772	#A2137 (includes AES-CBC, AES-CTR, AES-GCM)

Functions	Standards	Certificates
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	#A2137 (includes RSA SigGen (FIPS 186-4) and RSA SigVer (FIPS 186-4))
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, P-521 ISO/IEC 14888-3, Section 6.4	#A2137 (includes ECDSA SigGen (FIPS 186-4) and ECDSA SigVer (FIPS 186-4))
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	#A2137 (includes SHA-1, SHA2-256, SHA2-384, SHA2-512)
Keyed-hash message authentication (FCS_COP.1/KeyedHash)		
• HMAC-SHA-1 (block size 512 bits, key size 160 bits and digest size 160 bits) • HMAC-SHA-256 (block size 512 bits, key Size 256 bits and digest size 256 bits) • HMAC-SHA-384 (block size 1024 bits, key Size 384 bits and digest size 384 bits) • HMAC-SHA-512 (block size 1024 bits, key Size 512 bits and digest size 512 bits)	ISO/IEC 9797-2:2011	#A2137 (includes HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512)
Random bit generation (FCS_RBG_EXT.1)		
CTR_DRBG (AES) from a hardware-based noise source with one independent software-based noise source of 256 bits of non-determinism	ISO/IEC 18031:2011	#A2137 (includes Counter DRBG)

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 (2048/3072/4096-bit RSA, ECC with P-256/P-384/P-521, "safe-prime" Diffie-Hellman Group 14, and 2048-bit FFC). This section says that the TOE acts as both a sender and as a recipient for RSA/ECC/FFC schemes.

Upon review of section 6.2 and section 6.7 of [ST], it is clear that the schemes are used for the following functions:

- TLS: RSA/ECC/FFC (TOE is both sender and receiver) – note that RSA 2048 and ECDSA P-256/P-384/P-521 are supported for this.
- SSH: ECC/Diffie-Hellman group 14 (TOE is receiver) – note that ECDSA P-256/P-384/P-521 are supported for this.
- X.509: RSA/ECC (TOE is neither sender nor receiver since the cert request is exported outside the TOE and sent to a CA using a non-TOE mechanism) – note that RSA 2048/3072/4096 and ECDSA P-256/P-384 are supported for this.
- SSH (key pairs for authentication): RSA (TOE is neither sender nor receiver since it is for authentication and not protocol establishment) – note that RSA 2048/3072/4096 is supported for this.

Upon review of section 6.2 and section 6.3 of [ST], it is also clear that the RSA and ECC key generation schemes are used in support of key generation for X.509 certificate requests.

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 6.7.1 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

Key Generation for FIPS PUB 186-4 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 CAVP certifications verifying asymmetric key generation, as follows.

Algorithm	Tested Capabilities	Certificates
RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	Key Generation Mode: B.3.6 Properties: Modulo: 2048 Primality Tests: C.2 Properties: Modulo: 3072 Primality Tests: C.2 Properties: Modulo: 4096 Primality Tests: C.2 Public Exponent Mode: Fixed Fixed Public Exponent: 010001 Public Key Format: Standard	CAVP #A2137 RSA KeyGen (FIPS186-4)
ECC schemes using "NIST curves" P-256, P-384, P-521, that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	Curve: P-256, P-384, P-521 Secret Generation Mode: Testing Candidates	CAVP #A2137 ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4)
FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1	Capabilities: L: 2048 N: 256	CAVP #A2137 DSA KeyGen (FIPS186-4)

Modified per TD0580

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.

Modified per TD0580.

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

Scheme	SFR	Service
RSA	FCS_TLSS_EXT.1	Administration

ECDH	FCS_SSHC_EXT.1	Audit Server
ECDH	FCS_IPSEC_EXT.1	Authentication Server

The information provided in the example above does not necessarily have to be included as a table but can be presented in other ways as long as the necessary data is available.

Section 6.2 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 ECDHE (ECC), RFC 3526 safe-prime DH group 14 (FFC)
- TLS: FIPS 186-4 DHE (FFC), FIPS 186-4 ECDHE (ECC), FIPS 186-4 RSA

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

Modified per TD0580.

Key Establishment Schemes

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

SP800-56A Key Establishment Schemes

Performed in accordance with NIAP Policy Letter #5.

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

Algorithm	Tested Capabilities	Certificates
Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	Domain Parameter Generation Methods: P-256, P-384, P-521 Scheme: Ephemeral Unified: KAS Role: Initiator, Responder	CAVP #A2137 KAS-ECC-SSC
Finite field-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	Domain Parameter Generation Methods: FC Scheme: dhEphem: KAS Role: Initiator, Responder	CAVP #A2137 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.

Refer to test activities for FTP_TRP.1/Admin and FTP_ITC.1.

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

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 all keys stored in volatile memory are zeroized following their use and describes the zeroization method consistent with the claims made in FCS_CKM.4.1. Specifically, plaintext key data in volatile memory is overwritten with a random pattern followed by a read-verify, and plaintext key data in nonvolatile memory is overwritten with alternating ones and zeroes.

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 when FIPS-CC mode is enabled, all keys will be zeroized, including the KEK. To be in the evaluated configuration, 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 defined.

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 CAVP 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	CAVP #A2137 AES-CBC
AES as specified in ISO 18033-3, GCM as specified in ISO 19772	Direction: Decrypt, Encrypt IV Generation: Internal Key Length: 128, 256	CAVP #A2137 AES-GCM
AES as specified in ISO 18033-3, CTR as specified in ISO 10116	Direction: Decrypt, Encrypt Key Length: 128, 256	CAVP #A2137 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 6.6 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 CAVP certifications verifying digital signature services, as follows.

Algorithm	Tested Capabilities	Certificates
RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 4	**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	CAVP #A2137 RSA SigGen (FIPS186-4)

Algorithm	Tested Capabilities	Certificates
	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	
	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	CAVP #A2137 RSA SigVer (FIPS186-4)

Algorithm	Tested Capabilities	Certificates
ECDSA schemes using "NIST curves" P-256, P-384 and P-521 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5	**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	CAVP #A2137 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.

Sections 6.2 and 6.6 of [ST] indicate the hash function is associated with digital signature generation/verification and with HMAC. Additionally, sections 6.2 and 6.5 of [ST] state the TOE protects user passwords by storing hashed values of passwords using SHA-256.

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 6.7.1 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 CAVP certifications verifying cryptographic hashing, as follows.

Algorithm	Tested Capabilities	Certificates
SHS as defined in FIPS Pub 180-4	SHA-1 SHA-256 SHA-384 SHA-512	CAVP #A2137 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, block size, and digest size (which implicitly identifies the hash function 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 MAC algorithms are configured by default when FIPS-CC mode is enabled, but to further restrict them (e.g., use HMAC-SHA2-512 only), the administrator can use the 'set deviceconfig system ssh mac mgmt' CLI command.

2.2.7.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

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

Algorithm	Tested Capabilities	Certificates
HMAC that meets : FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code", and FIPS Pub 180-4, "Secure Hash Standard"	HMAC-SHA1 MAC: 160 Key Length: 256-2048 Increment 8 HMAC-SHA2-256 MAC: 256 Key Length: 256-2048 Increment 8 HMAC-SHA2-384 MAC: 384 Key Length: 256-2048 Increment 8 HMAC-SHA2-512 MAC: 512 Key Length: 256-2048 Increment 8	CAVP #A2137 HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512

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 no less than 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 CAVP 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	CAVP #A2137 Counter DRBG

2.2.9 FCS_SSHS_EXT.1 SSH Server Protocol

2.2.9.1 TSS Activities

FCS_SSHS_EXT.1.2 **Modified by TD0631** The evaluator shall check to ensure that the TSS contains a list of supported public key algorithms that are accepted for client authentication and that this list is consistent with signature verification algorithms selected in FCS_COP.1/SigGen (e.g., accepting EC keys requires corresponding Elliptic Curve Digital Signature algorithm claims). The evaluator shall confirm that the TSS includes the description of how the TOE establishes a user identity when an SSH client presents a public key or X.509v3 certificate. For example, the TOE could verify that the SSH client's presented public key matches one that is stored within the SSH server's authorized_keys file. If password-based authentication method has been selected in the FCS_SSHS_EXT.1.2, then the evaluator shall confirm its role in the authentication process is described in the TSS.

Section 6.2 of [ST] states that both public-key (RSA, consistent with FCS_COP.1/SigGen claims) and password-based authentication are supported, and that RSA is used for public-key authentication, consistent with FCS_SSHS_EXT.1.5. SSH authentication is for remote administrative users, per [ST] section 6.7.

FCS_SSHS_EXT.1.3
The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled.

Section 6.2 of [ST] states that packet data is accumulated in a buffer that contains 256KB. If the buffer space has been exhausted before the packet is complete (i.e. the packet is larger than 256KB), it will automatically be discarded by the TSF.

FCS_SSHS_EXT.1.4
The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

Section 6.2 of [ST] states that AES-CBC, AES-CTR, and AES-GCM, each with both 128- and 256-bit encryption, are all supported. It also states that no optional characteristics are specified.

FCS_SSHS_EXT.1.5
Modified by TD0631
The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the SSH server's host public key algorithms supported are specified and that they are identical to those listed for this component.

Section 6.2 of [ST] states that ssh-rsa is supported for public key authentication, consistent with FCS_SSHS_EXT.1.5. It also states that no optional characteristics are specified. This section also states that when RSA authentication is used, the administrator must associate a given user with the public portion of the RSA key the user is using for authentication.

FCS_SSHS_EXT.1.6
The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that the list corresponds to the list in this component.

Section 6.2 of [ST] states that HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512, and implicit integrity (when GCM is used as the encryption algorithm) are the TOE's supported data integrity algorithms, consistent with FCS_SSHS_EXT.1.6.

FCS_SSHS_EXT.1.7
The evaluator shall check the TSS to ensure that it lists the supported key exchange algorithms, and that the list corresponds to the list in this component.

Section 6.2 of [ST] states that diffie-hellman-group14-sha1, and EC-Diffie-Hellman with P-256, P-384, and P-521 are the TOE's only supported key exchange algorithms.

FCS_SSHS_EXT.1.8
The evaluator shall check that the TSS specifies the following:

a) Both thresholds are checked by the TOE.
b) Rekeying is performed upon reaching the threshold that is hit first.

Section 6.2 of [ST] states that the TOE maintains both time-based (configurable from 10 seconds to one hour) and traffic-based (configurable from 10 MB to 4000 MB) thresholds for rekeying, and that rekeying is performed after whichever threshold is reached first.

2.2.9.2 Guidance Activities

FCS_SSHS_EXT.1.4
The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Section 6.4 of [CCECG] describes how to configure the supported SSH encryption algorithms and lists the six specific algorithms that are required in order to conform to [ST].

FCS_SSHS_EXT.1.5
The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

[ST] states that ssh_rsa is the only supported public key algorithm. Section 6.6 of [CCECG] affirms this claim and describes how to enable public key authentication for SSH.

FCS_SSHS_EXT.1.6
The evaluator shall also check the guidance documentation to ensure that it contains instructions to the Security Administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed).

Section 6.4 of [CCECG] describes how to configure the supported data integrity algorithms using the 'set deviceconfig system ssh mac mgmt' command but also notes that enabling FIPS-CC mode will set the algorithms claimed by [ST] by default, so this command is only needed if further restrictions from that baseline are desired.

FCS_SSHS_EXT.1.7
The evaluator shall also check the guidance documentation to ensure that it contains instructions to the Security Administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections with the TOE.

The SSH key exchange algorithms are not separately configurable; enabling FIPS-CC mode as described in section 6.2 of [CCECG] is sufficient to ensure this function is configured properly.

FCS_SSHS_EXT.1.8
If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, then the evaluator shall check that the guidance documentation describes how to configure those thresholds. Either the allowed values are specified in the guidance documentation and must not exceed the limits specified in the SFR (one hour of session time, one gigabyte of transmitted traffic) or the TOE must not accept values beyond the limits specified in the SFR. The evaluator shall check that the guidance documentation describes that the TOE reacts to the first threshold reached.

Section 6.5 of [CCECG] describes how to configure the SSH rekey interval in both time and data amounts and states what the default values are for these functions when FIPS-CC mode is enabled.

2.2.9.3 Test Activities

FCS_SSHS_EXT.1.2
Modified by TD0631
Test 1: For each supported client public-key authentication algorithm, the evaluator shall configure a remote client to present a public key corresponding to that authentication method (e.g., 2048-bit RSA key when using ssh-rsa public key). The evaluator shall establish sufficient separate SSH connections with an appropriately configured remote non-TOE SSH client to demonstrate the use of all applicable public key algorithms. It is sufficient to observe the successful completion of the SSH Authentication Protocol to satisfy the intent of this test.

The TOE supports a single client public-key authentication algorithm (ssh-rsa). The evaluator configured a remote SSH client to present an ssh-rsa public key and confirmed the client was able to establish an SSH connection to the TOE using the ssh-rsa public key to authenticate.

FCS_SSHS_EXT.1.2
Modified by TD0631
Test 2: The evaluator shall choose one client public key authentication algorithm supported by the TOE. The evaluator shall generate a new client key pair for that supported algorithm without configuring the TOE to recognize the associated public key for authentication. The evaluator shall use an SSH client to attempt to connect to the TOE with the new key pair and demonstrate that authentication fails.

The TOE supports a single client public-key authentication algorithm (ssh-rsa). The evaluator generated a new ssh-rsa key pair on the remote SSH client without configuring the TOE to recognize the associated public key for authentication. The evaluator confirmed the client was unable to establish an SSH connection to the TOE using the new key pair.

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

The evaluator performed tests for the SSH interface with a password credential and confirmed that user authentication succeeded when the correct password was provided by the connecting SSH client.

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

The evaluator performed tests for the SSH interface with a password credential and confirmed that user authentication failed when an incorrect password was provided by the connecting SSH client.

FCS_SSHS_EXT.1.3
The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

The evaluator successfully authenticated to the TOE via an SSH client and sent a packet larger than 256K bytes to the TOE. The evaluator confirmed that the TOE rejected the packet and closed the connection.

FCS_SSHS_EXT.1.4

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

The evaluator successfully connected to the TOE via an SSH client and confirmed that the TOE offered only encryption algorithms claimed in the Security Target.

FCS_SSHS_EXT.1.5

Modified by TD0631

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

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

The testing performed in Test 1 for FCS_SSHS_EXT.1.2 above covers this test.

FCS_SSHS_EXT.1.5

Modified by TD0631

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

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

The evaluator configured a remote SSH client to allow it to authenticate only an ED25519 server host public key. The evaluator attempted to establish an SSH connection from the remote SSH client to the TOE and observed that the connection was rejected.

FCS_SSHS_EXT.1.6

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

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

The evaluator attempted to connect to the TOE via an SSH client using each of the MAC algorithms specified in the Security Target and verified that the connection succeeded.

FCS_SSHS_EXT.1.6

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

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

The evaluator attempted to connect to the TOE via an SSH client that only allowed hmac-md5 as the MAC algorithm and verified that the connection was denied.

FCS_SSHS_EXT.1.7

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

The evaluator attempted to connect to the TOE via an SSH client that only allowed Diffie-hellman-group1-sha1 key exchange and verified that the connection was denied.

FCS_SSHS_EXT.1.7

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

The evaluator attempted to connect to the TOE via an SSH client using each of the key exchange claimed in the Security Target and verified that the connection was successful.

FCS_SSHS_EXT.1.8

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

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

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

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

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

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

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

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

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

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

The evaluator configured the TOE's rekey threshold at 10MB and 1 hour independently. The evaluator connected to the TOE via an SSH client and ran separate tests confirming that when each threshold was met, the TOE performed a rekey.

2.2.10 FCS_TLSC_EXT.1 TLS Client Protocol without Mutual Authentication

2.2.10.1 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 user are relaxed and the identifier may also be established through a "Gatekeeper" discovery process. The TSS should 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] lists the supported reference identifiers as CN (subject), FQDN (hostname), and IPv4 addresses for all uses of TLS client communications. This section also states that wildcards are supported, that certificate pinning is not, and that the CN field is composed of IPv4 address data that conforms to RFC 3986. This section also describes the TOE's comparison of network address fields based on converting the IP address to machine byte order and then flipping to big-endian if it is not already in that format.

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 (RFC5952 for IPv6, RFC 3986 for IPv4) is enforced.

[ST] section 6.2 states that both TLS client interfaces support IP addresses in the CN as a reference identifier. This section states that this follows the rules described in RFC 3986 for IPv4. This section states that network addresses are presented in big-endian order and will be converted to machine byte order. If the machine byte order is little endian, it is flipped to big-endian and then compared.

FCS_TLSC_EXT.1.4
The evaluator shall verify that TSS describes the Supported Elliptic Curves/Supported Groups Extension and whether the required behaviour is performed by default or may be configured.

Section 6.2 of [ST] states that the TSF supports secp256r1, secp384r1, and secp521r1 by default.

2.2.10.2 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 identifiers as specified in RFC 6125 and IPv4 addresses in SAN or CN. Section 6.7.1 of [CCECG] provides detailed instructions on how to configure the reference identifier used to check the identity of an external syslog server. Section 6.7.2 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 RFC5280 attributes.

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

FCS_TLSC_EXT.1.4
If the TSS indicates that the Supported Elliptic Curves/Supported Groups Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Elliptic Curves/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].

2.2.10.3 Test Activities

For all tests in this chapter the TLS server used for testing of the TOE shall be configured not to require mutual authentication.

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 initiated connections from the TOE 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
Test 2: The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field, and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.

The evaluator confirmed that a TLS server presenting a certificate with the Server Authentication purpose in the extendedKeyUsage field resulted in a successful connection in Test 1 above. 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 did not accept the connection.

FCS_TLSC_EXT.1.1
Test 3: 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 did not accept the connection.

FCS_TLSC_EXT.1.1

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

a) The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.

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 did not accept the connection.

Test 4): Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.

The evaluator configured a remote TLS server to present a Server Hello message that contained a cipher suite that did not match any presented in the Client Hello handshake. The evaluator confirmed that the TOE did not accept the connection.

Test 4c) [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension the evaluator shall configure the server to perform an ECDHE or DHE key exchange in the TLS connection using a non-supported curve/group (for example P-192) and shall verify that the TOE disconnects after receiving the server's Key Exchange handshake message.

The evaluator configured a remote TLS server to present an ECDHE cipher suite specifying the non-supported P-192 curve. The evaluator confirmed that the TOE did not accept the connection.

FCS_TLSC_EXT.1.1

Test 5: The evaluator performs the following modifications to the traffic:

a) Change the TLS version selected by the server in the Server Hello to a non-supported TLS version and verify that the client rejects the connection.

The evaluator modified the TLS server to present a Server Hello message with a non-supported TLS version (1.0) and attempted a connection from the TOE. The evaluator confirmed that the TOE did not accept the connection.

Test 5) [conditional]: If using DHE or ECDH, modify the signature block in the Server's Key Exchange handshake message, and verify that the handshake does not finish successfully, and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.

The evaluator modified the TLS server to send a Server Key Exchange message with a modified signature block. The evaluator confirmed that the TOE did not accept the connection.

FCS_TLSC_EXT.1.1

Test 6: The evaluator performs the following 'scrambled message tests':

a) Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application flows.

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.

Test 6b): Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake does not finish successfully and no application data flows.

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.

Test 6c): 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.

The evaluator configured a TLS server to modify the nonce sent in the Server Hello handshake message and confirmed the TOE terminated the connection after receiving the Server Key Exchange message.

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 and a certificate with an invalid IPv4 address

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 extension and a bad IPv4 address in the SAN extension.

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 and IPv4 identifiers in the CN field.

FCS_TLSC_EXT.1.2

Test 4: 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 and IPv4 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):

1) [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.
2) [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 leftmost 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.)

1. The evaluator configured the TLS server to present a certificate with a DNS value of test.*.leidos.ate in the CN field. The evaluator attempted a connection from the TOE and verified that the TOE did not accept the connection.
2. 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 communicate to a server with no left-most label (tlss.ate) and two left-most labels (test.tlss.leidos.ate) independently and attempted a connection for each. The evaluator confirmed that the TOE did not accept either connection.

FCS_TLSC_EXT.1.2

Modified by TD0634

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.

Test 6 [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 192.168.1.20, 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).

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 6.

The evaluator configured the TLS server to present a certificate with an IP address of 172.16.0.* in the CN field. The evaluator attempted a connection from the TOE and verified that the TOE did not accept the connection.

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 modifies 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):

1) 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.
2) 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-atserialNumber=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.

3) 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.
4) 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

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 evaluation team conducted this test in conjunction with FCS_TLSC_EXT.1.1 Test 1.

FCS_TLSC_EXT.1.3

Test 2: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.

The evaluator attempted to establish a TLS connection from the TOE to a server presenting a certificate that did not chain to any of the TOE's trusted root CAs and verified the connection failed. The evaluator attempted to establish a TLS connection from the TOE to a server presenting an otherwise valid but expired certificate and verified the connection failed.

FCS_TLSC_EXT.1.3

Test 3 [conditional]: The purpose of this test 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.

This test is not applicable because the TOE does not support any administrative overrides.

FCS_TLSC_EXT.1.4

Test 1 [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension, the evaluator shall configure the server to perform an ECDHE or DHE (as applicable) key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.

The evaluator configured the TLS server to perform an ECDHE key exchange using the secp256r1 curve and attempted a connection from the TOE. The evaluator confirmed that the connection was successful. The evaluator repeated this test using the secp384 and secp521 curves.

2.2.11 FCS_TLSC_EXT.2 TLS Client support for mutual Authentication

2.2.11.1 TSS Activities

FCS_TLSC_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 X.509 certificates are used for both TLS client and server authentication.

2.2.11.2 Guidance Activities

FCS_TLSC_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.

[ST] section 6.2 states the TOE supports client-side mutual authentication for connections to external syslog servers. Section 6.7.1 of [CCECG] provides instructions for configuring the client-side certificate for connection to an external syslog server.

2.2.11.3 Test Activities

FCS_TLSC_EXT.2.1
For all tests in this chapter the TLS server used for testing of the TOE shall be configured to require mutual authentication.
FCS_TLSC_EXT.2.1
(covered by FCS_TLSC_EXT.1.1 Test 1 and testing for FIA_X.509_EXT.*).

Refer to testing conducted or FCS_TLSC_EXT.1.1 and FIA_X509_EXT.*.

2.2.12 FCS_TLSS_EXT.1 TLS Server Protocol without mutual authentication

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.

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.

FCS_TLSS_EXT.1.2
The evaluator shall verify that the TSS contains a description of how the TOE technically prevents the use of old SSL and TLS versions.

Section 6.2 of [ST] states that all TLS versions other than 1.1 and 1.2 are rejected, consistent with FCS_TLSS_EXT.1.2. This rejection is performed by default.

FCS_TLSS_EXT.1.3

Modified by TD0635

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

Section 6.2 of [ST] states that when an ECDHE or DHE ciphersuite is chosen, the TOE includes the relevant key agreement parameters (2048-bit Group 14 DHE, secp256r1 ECDHE) in the Server Key Exchange message as per RFC 5246.

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) and/or if session resumption based on session tickets is supported (RFC 5077).

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 session ID or tickets for a single context (no configuration needed).

Added per TD0569.

FCS_TLSS_EXT.1.4

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

[ST] section 6.2 states the TOE supports session resumption using session ID or 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.

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 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 6.7.4 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.

The parameters used for generation are a product of the cipher suite that is negotiated with the TLS client accessing the TOE. No separate configuration for this is required.

Added per TD0569.

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.

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 ciphersuite was successful.

FCS_TLSS_EXT.1.1
Test 2: The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the server denies the connection.

The evaluator configured a TLS client to request the TLS_NULL_WITH_NULL_NULL ciphersuite claimed in its Client Hello and attempted a connection to the TOE. The evaluator then attempted a connection to the TOE using no claimed ciphers in the Client Hello. The evaluator confirmed that the TOE did not accept either connection.

FCS_TLSS_EXT.1.1

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

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

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.

FCS_TLSS_EXT.1.1

Test 3b): (Test Intent: The intent of this test is to ensure that the server's TLS implementation immediately makes use of the key exchange and authentication algorithms to: a) Correctly encrypt (D)TLS Finished message and b) Encrypt every (D)TLS message after session keys are negotiated.)

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

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

The evaluator examined the packets that were captured during FCS_TLSS_EXT.1.1 Test 1 with the ciphersuite TLS_DHE_RSA_WITH_AES_128_CBC_SHA and verified that the Server's Encrypted Handshake Message was truly encrypted. The evaluator examined the frame number 6 and found that the data bytes for Encrypted Handshake Message did not contain "16 03 03 00 40 14 00 00 0c" but were truly encrypted.

FCS_TLSS_EXT.1.2

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

The evaluator configured a TLS client to send an SSLv2, SSLv3, and TLSv1 Client Hello to the TOE independently. The evaluator confirmed that the TOE did not negotiate a TLS connection with each attempt.

FCS_TLSS_EXT.1.3

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

a) The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single supported elliptic curve specified in the Elliptic Curves Extension. The Evaluator shall verify (though a packet capture

or instrumented client) that the TOE selects the same curve in the Server Key Exchange message and successfully establishes the connection.

The evaluator configured a TLS client to specify the 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.

FCS_TLSS_EXT.1.3
Test 1b): The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g. secp192r1 (0x13)) specified in RFC4492, chap. 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

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.1.3
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. 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).

The TOE supports DHE parameters of 256 bytes 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

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.

The TOE does not support RSA key establishment cipher suites, so this test is not applicable

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).

Modified per TD0569.

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

a) The client sends a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket.
b) The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake).

c) The client verifies the Server Hello message contains a zero-length session identifier or passes the following steps:
Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID.
d) The client completes the TLS handshake and captures the SessionID from the ServerHello.
e) The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d).
f) The client verifies the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

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 ID or session tickets. As such, this test is not applicable.

Modified per TD0569.
FCS_TLSS_EXT.1.4
Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

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

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.

The evaluator configured a TLS client to support only session IDs. The evaluator connected to the TOE and observed that the TOE was capable of resuming a session using a valid session ID. The evaluator then caused the TLS client to kill the TLS session and attempted to resume the session with a session ID that was not valid. The evaluator observed that instead of resuming the session the TOE performed a full exchange with a different session ID.

Modified per TD0556 and 0569.

FCS_TLSS_EXT.1.4

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

a) 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.**

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.

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

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 a modified session ticket to the TOE that the TOE rejected the session ticket and attempted to perform a full handshake.

2.2.13 FCS_TLSS_EXT.2 TLS Server support for Mutual Authentication

2.2.13.1 TSS Activities

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

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 describes the TOE's use of X.509 certificates for TLS client authentication.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
The evaluator shall verify the TSS describes how the TSF uses certificates to authenticate the TLS client. 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, which includes a validation of the identity using an FQDN identifier in the subject or SAN field, or an IPv4 or IPv6 address in the subject or SAN field.

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.2.3
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 RFC6125. 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. In addition, the TOE supports IPv4 and IPv6 addresses. The TOE extracts the IP address from the subject or SAN field in the certificate and matches it exactly to the referenced identifier.

2.2.13.2 Guidance Activities

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
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 6.7.2 of [CCECG] includes instructions for configuring the client-side certificates for TLS mutual authentication used by firewalls.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
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.2.3
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 ensures this description includes all types of identifiers described in the TSS and, if claimed, configuration of the TOE to use a directory server.

Section 6.7.2 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. In addition, the TOE supports IPv4 and IPv6 addresses. The TOE extracts the IP address from the subject or SAN field in the certificate and matches it exactly to the referenced identifier.

2.2.13.3 Test Activities

For all tests in this chapter the TLS client used for testing of the TOE shall support mutual authentication.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 1a [conditional]: If the TOE requires or can be configured to require a client certificate, the evaluator shall configure the TOE to require a client certificate and send a Certificate Request 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 supports fallback authentication functions and these functions cannot be disabled. The evaluator shall configure the fallback authentication functions on the TOE and configure the TOE to send a Certificate Request 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 the TOE authenticates the connection using the fallback authentication functions as described in the TSS.

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

The TOE does not support fallback authentication, so this test is not applicable.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 2 [conditional]: If TLS 1.2 is claimed for the TOE, the evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify that the connection is denied.

The evaluator configured a TLS client to present a client certificate signed using an algorithm not in the list presented by the TOE. The evaluator observed the TOE rejected the connection.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 3: The aim 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 carry out 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 4: The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that the server denies the connection. Ideally, the two certificates should be identical except for the Client Authentication purpose.

The evaluator configured a TLS client to present a client certificate that possessed only the Extended Key Usage of ClientAuthentication. The evaluator observed that the TOE accepted the certificate and the connection was completed successfully. The evaluator then configured the TLS client to present a client certificate that possessed only the Extended Key Usage of ServerAuthenticaiton. The evaluator observed that the TOE rejected the certificate and the connection was not completed.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 5: The evaluator shall perform the following modifications to the traffic:

a) 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.

This test is covered by testing performed in the first part of Test 4 for FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2.

Test 5b): 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 RFC5246 Sec 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.

The evaluator modified the last byte of the client's CertificateVerify handshake record and observed that the TOE rejected the connection after receiving the CertificateVerify message.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

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

Test 6: 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.

This test is covered by testing performed in the first part of Test 4 for FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

Test 7: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.

The evaluator attempted to establish a TLS connection from the TOE to a server presenting a certificate that did not chain to any of the TOE's trusted root CAs and verified the connection failed. The evaluator attempted to establish a TLS connection from the TOE to a server presenting an otherwise valid but expired certificate and verified the connection failed.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
Test 8 [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.

This test is not applicable because the TOE does not support any administrative overrides.

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 an expected identifier. The evaluator verified that the TOE rejected the 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.

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. 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 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 that users can be locked out for a configurable amount of time between 1 and 60 minutes. Alternatively, the lockout period can be configured to be indefinite, in which case an administrator must take action to manually unlock the offending account.

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 authentication mechanisms can be configured on a per-user basis and requires that at least one administrator be configured to use an SSH public key for authentication because the lockout mechanism only applies to password-based authentication.

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 describes how to unlock manually a locked account.

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's number of successive unsuccessful authentication attempts to 5 and the lockout time period to 5 minutes. The evaluator then attempted to authenticate to the TOE with an incorrect password 5 times and on the 6th time attempted to use the correct password. The evaluator confirmed that the TOE did not allow access. The evaluator then waited until just before 5 minutes had expired and again attempted to authenticate to the TOE with the correct password and verified access was denied. The evaluator then waited until after the 5 minute threshold had expired and attempted to authenticate to the TOE with the correct password and verified that access was granted.

The evaluator then configured the TOE to lockout an account with 6 failed authentication attempts and configured the lockout time to be 0, meaning an administrator was required to unlock the locked account. The evaluator then attempted to authenticate to the TOE with an incorrect password 6 times. After the 6 failed password attempts the evaluator presented the valid password and observed that the attempt failed. The evaluator observed that after 5 minutes the user could not log in with the valid password. The evaluator then authenticated to the TOE as an administrator that was not locked out and unlocked the user's account. The evaluator confirmed the user could now login with the correct password.

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 evaluator performed this test in conjunction with Test 1, as described above.

2.3.2 FIA_PMG_EXT.1 Password Management

2.3.2.1 TSS Activities

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

[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.

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 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 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 requirements and justify the subset of those characters chosen for testing.

The evaluator configured the minimum password length on the TOE to be 8 and attempted to set a password that was only 7 characters long. The evaluator observed that the TOE rejected the password change attempt. The evaluator then configured the minimum password length on the TOE to be 15 and attempted to set a password that was only 14 characters long. The evaluator observed that the TOE again rejected the password change attempt.

2.3.3 FIA_UAU_EXT.2 Password-based Authentication Mechanism

2.3.3.1 TSS Activities

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

2.3.3.2 Guidance Activities

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

2.3.3.3 Test Activities

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

2.3.4 FIA_UAU.7 Protected Authentication Feedback

2.3.4.1 TSS Activities

None defined.

2.3.4.2 Guidance Activities

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

Section 5.1.1 of [CCECG] states SSH is used for both local and remote administration. The TOE's Ethernet management port must be used, even when connecting locally, because once FIPS-CC mode is enabled, the console port is disabled. The administrator does not need to perform any preparatory steps to ensure authentication data is not revealed when entered during a local login attempt.

2.3.4.3 Test Activities

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

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

This test was performed in conjunction with FIA_AFL.1 Test 1, where the evaluator made a number of attempts to authenticate to the TOE via password and confirmed the TOE provided no feedback as to the characters that had been entered.

2.3.5 FIA_UIA_EXT.1 User Identification and Authentication

2.3.5.1 TSS Activities

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

Section 6.3 of [ST] states administrators can logon to the TOE's CLI using a secure connection from an SSH client. The TOE provides access to the locally via a direct RJ-45 Ethernet cable connection, and remotely using an SSHv2 client. The TOE supports username for identification and password (defined internal to the TOE) or SSH public key for authentication. The TOE provides a logon successful note when the user provides correct credentials matching a defined account on the TOE.

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

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. There is no distinction between the behavior of the TOE in the case of local versus remote authentication.

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

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 user identification and authentication. The description shall cover authentication and identification for local and remote TOE administration. For each TOE component that does not support authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 the TSS shall describe any unauthenticated services/services that are supported by the component.

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

2.3.5.2 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 6.6 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].

2.3.5.3 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 that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

This test was performed in conjunction with FIA_AFL.1 Test 1, which shows correct and incorrect authentication using username and password, and with FCS_SSHS_EXT.1.2 Tests 1 and 2, which show correct and incorrect authentication using username and public key authentication.

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 before 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.

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 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.

The TOE utilizes SSH with no networking equipment connected between the client and TOE to accomplish local access. Hence, this test is covered by FIA_UIA_EXT.1, Test 2.

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

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

2.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

2.3.6.1 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). It is not sufficient to verify the status of a X.509 certificate only when it's loaded onto the device. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).

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).

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 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 revocation checking is used when checking the revocation status of presented TLS peer certificates. No difference in the handling of certificate chains vs leaf certificates is identified.

2.3.6.2 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 6.7 of [CCECG] states the check of validity of certificates takes place during the TLS handshake.

Section 6.7.1 of [CCECG] states the TOE automatically checks revocation status based on CRL information located in the certificate for Syslog connections.

Section 6.7.2 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.

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.

2.3.6.3 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 not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).

The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev:

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 certificate to be used in the function, and shall use this chain to demonstrate that the function succeeds.

The evaluator conducted this test in conjunction with FCS_TLSS_EXT.1.1 Test 1 and FCS_TLSS_EXT.1.1 Test 1.

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 conducted this test in conjunction with Test 2 for FCS_TLSC_EXT.1.3 and Test 7 for FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2.

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

The evaluator conducted this test in conjunction with Test 2 for FCS_TLSC_EXT.1.3 and Test 7 for FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2.

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

The evaluator attempted a TLS connection with the TOE using both revoked and non-revoked end-entity certificates and Intermediate CA certificates to be checked via CRL. The evaluator confirmed that non-

revoked certificates resulted in a successful connection and revoked certificates resulted in the TOE denying the connection. The evaluator performed this test for both the TOE's TLS client and TLS server (with mutual authentication).

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

The evaluator configured a TLS peer to use a certificate that contained revocation information and chained to a CA that lacked the CRLsign EKU. The evaluator verified that the TOE rejected the connection attempt and did not establish a connection. The evaluator performed this test for both the TOE's TLS client and TLS server (with mutual authentication).

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.

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

The evaluator confirmed that the TOE would not establish a TLS connection with a certificate modified in the last byte.

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.

Added per TD0527

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

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

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 used 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 difference 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 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 it to verify that the TOE accepts only certificates that have been marked as CA certificates by using basicConstraints with the CA flag set to True (and implicitly 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 confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

The evaluator attempted to load a CA certificate onto the TOE that did not contain the basicConstraints extension and verified that the TOE did not accept the certificate.

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

The evaluator attempted to import into the TOE a certificate that had the BasicConstraints value set to FALSE. The evaluator observed that the TOE allowed the certificate to be imported and treated the certificate the same as a regular end-entity certificate. Next, the evaluator attempted to use the imported certificate with the BasicConstraints value set to FALSE as a CA certificate in a certificate profile and observed that the TOE did not recognize the certificate as a CA that could be configured as a CA certificate to validate other certificates.

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 TOE uses certificates only for authentication of TLS connections, so no further testing is necessary.

2.3.7 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.

2.3.7.1 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.

The evaluator shall examine the TSS to confirm that it describes the behavior 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.

2.3.7.2 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 6.7 of [CCECG] provides guidance to the administrator configuring secure connections with a syslog server (section 6.7.1 of [CCECG]) and firewall devices (section 6.7.2 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 6.7.1 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 6.7.2 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.

2.3.7.3 Test Activities

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

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 checking 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 and did not allow a connection to complete.

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 information and observed that the TOE rejected the connection. Next, the evaluator configured the TOE to allow the connection if the TOE could not verify the certificate's revocation status during the connection attempt. The evaluator caused a TLS client to attempt a connection to the TOE while presenting a certificate containing revocation status information and observed that the TOE accepted the connection.

2.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

2.3.8.1 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.

2.3.8.2 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 Message. 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 certificate request message.

Section 6.7.1 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.

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

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

General requirements for distributed TOEs

Guidance Documentation

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

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. (AAR Section 2.4). 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

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. The evaluator also observed that the attempt to update the TOE without user administrative privileges failed as there was no way to log onto the TOE without a valid Security Administrator account.

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.

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

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

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 TSF behavior, which includes "managing X.509v3 certificates in the trust store."

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 locally and remotely – [CCECG] section 5.1.1 and [Admin] under "Configure the WildFire Appliance" (for instructions on how to access the TOE locally and remotely) 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 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 authentication failure parameters for FIA_AFL.1 – [CCECG] section 7.6
- 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), 6.4 (SSH configuration), 6.6 (SSH configuration), 6.7 and subsections (configuration for specific TLS logical interfaces)
- Ability to set the time which is used for time-stamps – [CCECG] section 7.4
- Ability to re-enable an Administrator account – [CCECG] section 7.6
- Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors – [CCECG] section 6.7.1 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 import X.509v3 certificates to the TOE's trust store – [CCECG] section 6.7.1
- Ability to manage the trusted public keys database – [CCECG] section 6.6.

[Admin], under "Configure the WildFire Appliance", defines the two default administrator roles as 'superuser', which has read/write access to the entire system, and 'super reader', which has read-only access to the entire system. New users can be assigned to these roles. It is clear from the description of these roles what privileges they grant and what users should be assigned what roles to ensure that their level of access to the TSF is appropriate. In particular, the 'superuser' role clearly corresponds to the Security Administrator role defined by [NDcPP].

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 6.7.2 and 7.2 of [CCECG] describe the process for loading certificates, including CA certificates. The 'request certificate generate' command described in section 6.7.1 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.

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 locally and remotely 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. All CLI functionality is identical regardless of whether the TOE is accessed locally or remotely.

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.

Section 6.4 of [ST] states the TOE provides a CLI to support security management of the TOE. The CLI is accessible via direct connection to the management port on the device (local access), or remotely over SSHv2. The local interface supports the use of a dedicated Ethernet port that only supports communication with a whitelisted local IP address. Regardless of whether the physical interface is local or remote, the logical interface used to manage the TOE is through SSH CLI.

Section 5.1 of [CCECG] describes how to connect to the appliance locally using the Ethernet management port. This section appropriately warns the administrator to ensure the local connection is via the Ethernet management port. Section 6.1 of [CCECG] describes how to restrict management access by whitelisting for local IP addresses.

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 behavior 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.

2.4.3.2 Guidance Activities

See section 2.4.4.1 [of [ND-SD]].

These activities are addressed in section 2.4.3.1 above.

2.4.3.3 Test Activities

The evaluator tests management functions as part of testing the SFRs identified in section 2.4.4. (AAR Section 2.4.7) 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 locally and remotely;
  Tested in conjunction with FMT_SMR.2
- Ability to configure the access banner;
  Tested in conjunction with FTA_TAB.1
- Ability to configure the session inactivity time before session termination or locking;
  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 authentication failure parameters for FIA_AFL.1;
  Tested in conjunction with FIA_AFL.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_SSHS_EXT.1, FCS_TLSC_EXT.1, and FCS_TLSS_EXT.1
- Ability to re-enable an Administrator account
  Tested in conjunction with FIA_AFL.1
- 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.3
- Ability to import X.509v3 certificates to the TOE's trust store;
  Tested in conjunction with FIA_X509_EXT.3
- Ability to manage the trusted public keys database;
  Tested in conjunction with FCS_SSHS_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.

[ST] section 6.4 identifies the pre-defined superuser and read-only roles supported by the TOE and states that an 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.

The "Configure the WildFire Appliance" section of [Admin] describes first-time usage of the CLI via local access. 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; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.

All configuration activities and management is done through either a local or remote SSH session and issuing CLI commands to the TOE. The evaluator performed this testing in conjunction with FCS_SSHS_EXT.1 and throughout overall testing where administrative actions were performed (See FMT_SMF.1).

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.

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.

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.

As stated above, [ST] describes how password data is unavailable to user access by design.

2.5.1.2 Guidance Activities

None defined.

2.5.1.3 Test Activities

None defined.

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

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

2.5.2.2 Guidance Activities

None defined.

2.5.2.3 Test Activities

None defined.

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 TOE uses time data for auditing, cryptography, certificate validity checking, admin session timeouts, and time-based lockout following authentication failure. Time data is reliable through use of an internal hardware-based real-time clock.

2.5.3.2 Guidance Activities

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

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.

Following text added in accordance with TD0632:

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

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 uses 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 time on the TOE, then attempted to change the time. The evaluator queried the time again and verified that the time successfully changed.

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

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

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.

Modified by TD0632

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

This test is not applicable because the TOE does not obtain time from an underlying VS.

2.5.4 FPT_TST_EXT.1 TSF Testing

2.5.4.1 TSS Activities

The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

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 self-tests are run.

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], chap. 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. Note that the testing is not restricted to the cryptographic functions of the TOE.

b) [FIPS 140-2], chap. 4.9.1, Cryptographic algorithm test for the verification of the correct operation of cryptographic functions. Alternatively, national requirements of any CCRA member state for the security evaluation of cryptographic functions should be considered as appropriate.

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.

The evaluator confirmed that the TOE performs the identified self-tests during start-up.

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

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 should examine the guidance documentation instead.

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

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

[ST] does not claim a hash mechanism for verifying software updates so this activity is N/A.

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 needs to 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 or verification of published hash). The description shall include the procedures for successful and unsuccessful verification. The description shall correspond to the description in the TSS.

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.

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

[ST] does not claim the use of a published hash so this evaluation activity is N/A.

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

The evaluator checked the TOE's version, ran an update and then checked the 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 first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the

illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates:

1) A modified version (e.g. using a hex editor) of a legitimately signed update
2) An image that has not been signed
3) An image signed with an invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate signature)
4) If the TOE allows a delayed activation of updates the TOE must be able to display both the currently executing version and most recently installed version. The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify, that both, current version and most recently installed version, reflect the same version information as prior to the update attempt.

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

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

1) The evaluator obtains or produces an illegitimate update such that the hash of the update does not match the published hash. The evaluator provides the published hash value to the TOE and calculates the hash of the update either on the TOE itself (if that functionality is provided by the TOE), or else outside the TOE. The evaluator confirms that the hash values are different, and attempts to install the update on the TOE, verifying that this fails because of the difference in hash values (and that the failure is logged). Depending on the implementation of the TOE, the TOE might not allow the Security Administrator to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE
2) The evaluator uses a legitimate update and tries to perform verification of the hash value without providing the published hash value to the TOE. The evaluator confirms that this attempt fails. Depending on the implementation of the TOE it might not be possible to attempt the verification of the hash value without providing a hash value to the TOE, e.g. if the hash value needs to be handed over to the TOE as a parameter in a command line message and the syntax check of the command prevents the execution of the command without providing a hash value. In that case the mechanism that prevents the execution of this check shall be tested accordingly, e.g. that the syntax check rejects the command without providing a hash value, and the rejection of the attempt is regarded as sufficient verification of the correct behaviour of the TOE in failing to verify the hash. The evaluator then attempts to install the update on the TOE (in spite of the unsuccessful hash verification) and confirms that this fails. Depending on the implementation of the TOE, the TOE might not allow to even attempt updating the TOE after the verification of the

hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE

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

This test is not applicable because the TOE does not perform verification of a published hash.

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

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

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

Test 3 is skipped because the TOE does not use a published hash for update verification.

The only update mechanism is manual update, so all testing above was performed using that mechanism.

The TOE is not distributed so multiple iterations of testing for separate TOE components is not applicable.

2.6 TOE Access (FTA)

2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

2.6.1.1 TSS Activities

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

[ST] section 6.6 describes the session termination behavior, specifically that the same mechanism applies to both local and remote sessions. 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 states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period.

Section 7.6 of [CCECG] states the administrator can configure an idle session timeout for CLI users that access the TOE locally. 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

Test 1: The evaluator follows the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes

a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.

The TOE's console port is disabled in CC-FIPS mode, thus the Ethernet port was used for both local access and remote access. The network cable was plugged directly from the endpoint to the TOE for local access and through a switch for remote access. SSH is used to connect to the TOE for both local and remote access. For this reason, this test was performed in conjunction with FTA_SSL.3 Test 1.

2.6.2 FTA_SSL.3 TSF-initiated Termination

2.6.2.1 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, specifically that the same mechanism applies to both local and remote sessions. The inactivity time period is a configurable value between 1 and 1,440 minutes, with a default of 60 minutes.

2.6.2.2 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).

2.6.2.3 Test Activities

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

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

The evaluator configured the TOE to have inactivity timeout values of 5 and 10 minutes. The evaluator then authenticated to the TOE and verified when the inactivity value was hit the evaluator was logged out and had to re-authenticate to the TOE.

2.6.3 FTA_SSL.4 User-initiated Termination

2.6.3.1 TSS Activities

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

[ST] section 6.6 states users can enter the "exit" command to terminate local and remote user sessions.

2.6.3.2 Guidance Activities

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

Section 5.1.2 of [CCECG] states the 'exit' command is used to terminate an interactive CLI session (for both local and remote sessions).

2.6.3.3 Test Activities

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

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

The TOE's console port is disabled in CC-FIPS mode and thus the Ethernet port was used for both local access and remote access. The network cable was plugged directly from the endpoint to the TOE for local access and through a switch to for remote access. SSH is used to connect to the TOE for both local and remote access. For this reason, this test was performed in conjunction with FTA_SSL.4 Test 2.

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

The evaluator logged in 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 gain access to the TOE.

2.6.4 FTA_TAB.1 Default TOE Access Banners

2.6.4.1 TSS Activities

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

Section 6.6 of [ST] states that the TOE displays a configurable warning banner on the CLI prior to administrator authentication, regardless of whether local or remote access is being attempted.

2.6.4.2 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.

2.6.4.3 Test Activities

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

with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

The evaluator configured the TOE access banner and confirmed it 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 6.7 (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.

Evidence for this can be seen in FCS_TLSC_EXT.1.1 Test 1 for the TOE communicating as a TLS client and FCS_TLSS_EXT.1.1 Test 1 testing for the TOE communicating as a TLS server.

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.

This test was performed in conjunction with FTP_ITC.1, Test 4.

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

Evidence for this can be seen in FCS_TLSC_EXT.1.1 Test 1 for the TOE communicating as a TLS client and FCS_TLSS_EXT.1.1 Test 1 testing for the TOE communicating as a TLS server, where testing shows application data being communicated within the TLS v1.2 channel and no channel data being sent in plain text.

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

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

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

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

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, ~15 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.

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.

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.

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.

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 ensures 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 programing 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). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. The intent is that these

interfaces will be adequately tested, and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied.

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

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

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

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

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_TLSC_EXT.2, 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 (both local and remote) 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 Evaluation Activities in this section are described under the traditionally separate AGD families, the mapping between real TOE documents 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.

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.

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 evaluators reviewed Palo Alto's website and identified the TechDocs portal at https://docs.paloaltonetworks.com/. The evaluators observed that the "Products" dropdown menu included a link to WildFire, which includes release information and other product documentation, including [Admin]. [Admin] includes a section "Set Up and Manage a WildFire Appliance", with subsection "Configure the WildFire Appliance" that includes a link to the Palo Alto TechDocs portal, where the administrator can view and download [HRG]. As per NIAP policy, [CCECG] is 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 WF-500 appliance. [CCECG] (in section 1.2) and both [Admin] and [HRG] refer explicitly to the WF-500 appliance; all other versions of WildFire are cloud-based.

[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, and does not describe any external components or interfaces outside the scope of what [ST] claims.

3.3.1.3 AGD_OPE.1 Evaluation Activity

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

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

Modified per TD0536.

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

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

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

1) 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).

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

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

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).

3.3.1.6 Evaluator Actions for Assessing the Guidance Documentation (for distributed TOE)

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

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 [HRG] 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 [HRG] 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 [HRG] 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 to manage the security of the TSF as a product and as a component of the larger operational environment.

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.

The following are the ALC_CMC.1 CEM work units:

The evaluator shall check that the TOE provided for evaluation is labelled with its reference.

The evaluator compared the TOE references displayed on the CLI and TOE chassis with the hardware and software references provided in [ST] and guidance documentation and confirmed all references were consistent. Specifically, the evaluator verified that the TOE's hardware was identified using the model name defined in [ST], and that the software version was consistent with [ST]'s identification of it.

The evaluator shall check that the TOE references used are consistent.

The evaluator compared the TOE references displayed on the CLI and TOE chassis with the hardware and software references provided in [ST] and guidance documentation and confirmed all references were consistent. Specifically, the evaluator verified that the hardware model was identified using the model name defined in [ST], and that the software version was consistent with the operational guidance and with the ST's definition of the software version. The following screenshots and photographs demonstrate proper identification of the TOE.

TOE chassis (“WF-500” visible on lower right):

CLI identifying WF-500 model and software version 10.1.6-h4:

```
admin@PA-WF-500-2> show system info

hostname: PA-WF-500-2
ip-address: 172.16.13.45
public-ip-address: unknown
netmask: 255.255.0.0
default-gateway: 172.16.0.1
ip-assignment: static
mac-address: 0c:c4:7a:12:7c:ce
vm-interface-ip-address: unknown
vm-interface-netmask: unknown
vm-interface-default-gateway: 192.168.2.254
vm-interface-dns-server: 192.168.2.254
time: Tue Jul  5 17:49:58 2022
uptime: 0 days, 0:05:37
family: m
model: WF-500
serial: 009707000480
cloud-mode: non-cloud
sw-version: 10.1.6-h4
wf-content-version: 2039-2252
wf-content-release-date: 2022/07/01 11:00:01
logdb-version: 10.1.2
platform-family: m
operational-mode: fips-cc
device-certificate-status: None

admin@PA-WF-500-2>
```

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.

The following are the ALC_CMS.1 CEM work units:

The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

The required evaluation evidence comprises the information in [ST] coupled with the guidance documentation. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the guidance (see ALC_CMC.1-2), the evaluator implicitly confirmed the information required by this assurance component. Specifically, the evaluator verified that the TOE was included in the configuration list as well as the guidance documentation that is provided to end users of the TOE.

The configuration list shall uniquely identify the configuration items.

The required evaluation evidence comprises the information in the [ST] coupled with the guidance documentation. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the guidance (see ALC_CMC.1-2), the evaluator implicitly confirmed the information required by this assurance component. Specifically, the evaluator observed that each document is uniquely identified by name and version number and/or publication date.

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 should consult Appendix 709 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 A.9.3.1.

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 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 from May 24, 2022, to July 13, 2022.

3.6 Class AVA: Vulnerability Assessment

3.6.1 AVA_VAN.1 Vulnerability Survey

Vulnerability search activities modified per TD0564

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

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)

Modified per TD0547.

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

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

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

b) a mapping of the auditable events recorded by each distributed TOE component as in [NDcPP, 6.3.3]

c) additional information in the Preparative Procedures as identified in the refinement of AGD_PRE.1 in additional information in the Preparative Procedures as identified in 3.4.1.2 and 3.5.1.2.

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.

3.6.1.2 AVA_VAN.1 Evaluation Activity

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

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://nvd.nist.gov/).
- Vulnerability Notes Database (http://www.kb.cert.org/vuls/)
- Palo Alto Networks Security Advisories (https://security.paloaltonetworks.com/).

Specifically, the following search terms were used in these searches:

- Intel xeon e5-2620 (TOE processor)
- Sandy bridge (processor microarchitecture)
- PAN-OS 10.1 (TOE software platform)
- WildFire 10.1 (TOE software)
- Palo Alto Wildfire (vendor and product)
- Palo Alto Networks Wildfire (vendor and product variation)
- WF-500 (TOE hardware).

The evaluators performed these searches on 13 July 2022 and repeated them on 3 August 2022.

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].