Assurance Activities Report

for

Trend Micro TippingPoint Threat Protection System (TPS) v6.3

Version 1.0

6 December 2024

Prepared by:

leidos

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:

Trend Micro
11305 Alterra Parkway
Austin, TX 78758

The TOE Evaluation was Sponsored by:

Trend Micro
11305 Alterra Parkway
Austin, TX 78758

Evaluation Personnel:

Anthony Apted
Dawn Campbell
Josh Marciante
Armin Najafabadi
Allen Sant
Srilekha Vangala

Common Criteria Version:

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.

Common Evaluation Methodology Version:

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

Protection Profiles:

- *collaborative Protection Profile for Network Devices*, Version 2.2e, 23 March 2020 [CPP_ND_V2.2E].

Revision History

Version	Date	Description
0.1	12 February 2024	Initial draft.
0.2	28 February 2024	Updated for corrected ST and Guidance.
1.0	6 December 2024	Final version for Check-out.

Contents

1 Introduction

This document presents results from performing evaluation activities associated with the Trend Micro TippingPoint Threat Protection System (TPS) v6.3 evaluation. 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 *Evaluation Activities for Network Device cPP*, Version 2.2, December 2019 ([CPP_ND_V2.2-SD]), and including the following optional and selection-based SFRs: FAU_STG.1; FAU_STG_EXT.3/LocSpace; FCS_SSHC_EXT.1; FCS_SSHS_EXT.1; and FMT_MOF.1/Functions.

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 verified that the claimed NIST validation complies with the NIAP-approved PP requirements the TOE claims to satisfy. The CCTL verification of the NIST validation constitutes 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 [CPP_ND_V2.2E]. Rationale is included for those Technical Decisions that do not apply to this evaluation.

TD0527 Updates to Certificate Revocation Testing (FIA_X509_EXT.1)

This TD is not applicable to the TOE. It affects the cPP's iterations of FIA_X509_EXT.1, neither of which is claimed by the ST.

TD0528 NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4

This TD is not applicable to the TOE. It affects FCS_NTP_EXT.1, which is not claimed by the ST.

TD0536 NIT Technical Decision for Update Verification Inconsistency

This TD is applicable to the TOE.

TD0537 NIT Technical Decision for Incorrect Reference to FCS_TLSC_EXT.2.3

This TD is not applicable to the TOE. It affects FCS_TLSC_EXT.2, which is not claimed by the ST.

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

This TD is not applicable to the TOE. It affects FCS_DTLSC_EXT.1, which is not claimed by the ST.

TD0547 NIT Technical Decision for Clarification on developer disclosure of AVA_VAN

This TD is applicable to the TOE.

TD0555 NIT Technical Decision for RFC Reference incorrect in TLSS Test

This TD is not applicable to the TOE. It affects FCS_TLSS_EXT.1, which is not claimed by the ST.

TD0556 NIT Technical Decision for RFC 5077 question

This TD is not applicable to the TOE. It affects FCS_TLSS_EXT.1, which is not claimed by the ST.

TD0563 NIT Technical Decision for Clarification of audit date information

This TD is applicable to the TOE.

TD0564 NIT Technical Decision for Vulnerability Analysis Search Criteria

This TD is applicable to the TOE.

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

This TD is not applicable to the TOE. It affects FCS_TLSS_EXT.1 and FCS_DTLSS_EXT.1, neither of which are claimed by the ST.

TD0570 NIT Technical Decision for Clarification about FIA_AFL.1

This TD is applicable to the TOE.

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

This TD is applicable to the TOE.

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

This TD is applicable to the TOE.

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

This TD is applicable to the TOE.

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

This TD is applicable to the TOE.

TD0591 NIT Technical Decision for Virtual TOEs and hypervisors

This TD is applicable to the TOE.

TD0592 NIT Technical Decision for Local Storage of Audit Records

This TD is applicable to the TOE.

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

This TD is applicable to the TOE.

TD0632 NIT Technical Decision for Consistency with Time Data for vNDs

This TD is applicable to the TOE.

TD0635 NIT Technical Decision for TLS Server and Key Agreement Parameters

This TD is not applicable to the TOE. It affects FCS_TLSS_EXT.1, which is not claimed by the ST.

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

This TD is applicable to the TOE.

TD0638 NIT Technical Decision for Key Pair Generation for Authentication

This TD is applicable to the TOE.

TD0639 NIT Technical Decision for Clarification for NTP MAC Keys

This TD is not applicable to the TOE. It affects FCS_NTP_EXT.1, which is not claimed by the ST, as well as dependencies on the NTP claim which the ST does not make.

TD0670 NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing

This TD is not applicable to the TOE. It affects FCS_TLSC_EXT.2, which is not claimed by the ST.

TD0738 NIT Technical Decision for Link to Allowed-With List

This TD is applicable to the TOE.

TD0790 NIT Technical Decision: Clarification Required for testing IPv6

This TD is not applicable to the TOE. It affects FCS_DTLSC_EXT.1 and FCS_TLSC_EXT.1, neither of which are claimed by the ST.

TD0792 NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR

This TD is applicable to the TOE.

TD0800 Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance

This TD is not applicable to the TOE. It affects FCS_IPSEC_EXT.1, which is not claimed by the ST.

1.2 SAR Evaluation

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

SAR	Verdict
ASE_CCL.1	Pass
ASE_ECD.1	Pass
ASE_INT.1	Pass
ASE_OBJ.1	Pass
ASE_REQ.1	Pass
ASE_SPD.1	Pass
ASE_TSS.1	Pass
ADV_FSP.1	Pass
AGD_OPE.1	Pass
AGD_PRE.1	Pass
ALC_CMC.1	Pass
ALC_CMS.1	Pass
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.

1.3 Evidence

[ST] *Trend Micro TippingPoint Threat Protection System (TPS) v6.3 Security Target*, Version 1.0, September 23, 2024

[CCECG] *Common Criteria Evaluated Configuration Guide (CCECG) for TPS v6.3*, Document Version 1.0, September 2024

[HSIG] *Trend Micro TippingPoint Threat Protection System (TPS) Hardware Specification and Installation Guide*, April 2024

[CLI] *Trend Micro TippingPoint Threat Protection System (TPS) Command Line Interface Reference*, April 2024

[vTPSUG] *Trend Micro TippingPoint Virtual Threat Protection System (vTPS) User Guide*, April 2024

2 Security Functional Requirement Evaluation Activities

This section describes the evaluation activities associated with the SFRs defined in the ST that were taken from [CPP_ND_V2.2E] and the results of those activities as performed by the evaluation team. The evaluation activities are derived from [CPP_ND_V2.2-SD] and modified by applicable NIAP Technical Decisions. Evaluation activities for SFRs not claimed by the TOE have been omitted.

2.1 Security Audit (FAU)

2.1.1 Audit Data Generation (FAU_GEN.1)

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 6.1.1 of [ST] (“FAU_GEN.1: Audit Data Generation”) states when a cryptographic key is changed, the associated username and categorization of it as an SSH key is logged.

> For distributed TOEs the evaluator shall examine the TSS to ensure that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components. The evaluator shall ensure that this mapping of audit events to TOE components accounts for, and is consistent with, information provided in Table 1, as well as events in Tables 2, 4, and 5 (where applicable to the overall TOE). This includes that the evaluator shall confirm that all components defined as generating audit information for a particular SFR should also contribute to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component cover all the SFRs that it implements.

The TOE is not distributed so this 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 2.5.1 of [CCECG] (“Audit Events”) contains two tables (“Table 3 – Sample Audit Records” and “Table 4 – Sample Audit Records of Administrative Actions”) that together provide examples of each auditable event required by FAU_GEN.1.

The evaluator examined the audit record tables in [CCECG] and compared the contents to the audit record requirements specified in FAU_GEN.1.1, comprising:

- Start-up and shut-down of the audit functions (FAU_GEN.1.1a))
- All administrative actions listed in FAU_GEN.1.1c)
- Specifically defined auditable events listed in Table 3 of [ST] (“Auditable Events”).

The evaluator confirmed an example of each required audit record was included and that each record contained the required information of date and time the event was generated, the event type, the subject identity, the outcome of the event, and the additional audit record content specified in Table 3 of [ST], where applicable.

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 evaluator examined the supplied guidance documentation, identifying all mechanisms available to the administrator for configuring and managing the capabilities of the TOE. Those mechanisms related to the SFRs specified in the ST were identified and mapped to the applicable SFRs. In addition, the evaluator sought to confirm that all SFRs that would be expected to have a management capability related to them had appropriate management capabilities identified in the guidance documentation. Finally, the evaluator confirmed the identified administrative actions addressed the requirements of FAU_FEN.1.1c) (which provides a list of auditable administrative actions) and Table 4 of [CCECG].

The administrative actions identified as auditable (per FAU_GEN.1.1c)) are:

- Administrative login and logout (also covered by auditing of FIA_UIA_EXT.1)
- Changes to TSF data related to configuration changes (covered by auditing of FIA_AFL.1, FMT_MOF.1/ManualUpdate, FMT_SMF.1, FTM_STM_EXT.1, and FPT_TUD_EXT.1), and comprising:
    - Configuration of syslog export settings—referenced in [CLI] under "SSH configuration"
    - Configuring the cryptographic functionality (i.e., Enabling FIPS mode)—referenced in [CLI] under "Root commands > fips-mode-enable"
    - Set the date and time—referenced in [CLI] under "Root commands > date"
    - Setting maximum login attempt limit—referenced in [CLI] under "Edit running configuration commands > Contexts and related commands > running-aaa Context Commands > ips{running-aaa}login" (`ips{running-aaa}login maximum-attempts LOGINATTEMPTS`)
    - Setting lockout period following excessive login failures—referenced in [CLI] under "Edit running configuration commands > Contexts and related commands > running-aaa Context Commands > ips{running-aaa}login" (`ips{running-aaa}login lockout-period DURATION`)
    - Configuration of size of audit record storage—referenced in [CLI] under "Log configure commands > rotate"
    - Specifying the inactivity time period—referenced in [CLI] under "Edit running configuration commands > Contexts and related commands > running-aaa Context Commands > ips{running-aaa}login" (`ips{running-aaa}login cli-inactive-timeout (MINUTES)`)
    - Configuring the banner displayed prior to authentication—referenced in [CLI] under "Edit running configuration commands > Contexts and related commands > running-aaa Context Commands > ips{running-aaa}login-banner"
    - Initiating manual update—referenced in section 2.9 of [CCECG] ("TOE Updates")

    - Management of the trusted public keys database—referenced in sections 2.6.2.1, 2.6.2.2, and 2.6.2.3 of [CCECG] ("SSH Host Key Configuration", "SSH Client Private Key Configuration", and "SSH User Public Key Configuration" respectively).

- Generating/import of, changing, or deleting of cryptographic keys
- Resetting passwords (referenced in [CLI] under "Root commands > chpasswd").

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 verified that the TOE successfully generated each of the audit records which are required for each activity specified by the FAU_GEN.1 SFR.

> 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.

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

2.1.2 User Identity Association (FAU_GEN.2)

2.1.2.1 TSS & 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.

2.1.2.2 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.

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

2.1.3 Protected Audit Trail Storage (Audit Data) (FAU_STG.1/Audit)

2.1.3.1 TSS Activities

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

Section 6.1.3 of [ST] ("FAU_STG.1: Protected Audit Trail Storage") states audit data stored locally by the TOE is stored in two files—the "System" log, and the "Audit" log. The System log records information about the software processes that control the TOE, including startup and shutdown of the audit function. The Audit log records all other required audit events as specified in FAU_GEN.1.

Section 6.1.4 of [ST] ("FAU_STG_EXT.1: Protected Audit Event Storage") states each TOE device (including vTPS) allocates approximately one eighth of its internal disk space for local storage of audit records. Devices that support less than 5 Gbps inspection throughput have 8 GB internal disk space while devices that support 5 Gbps and above inspection throughput have 32 GB internal disk space. This implies approximately 1 GB audit log file disk space on devices with less than 5 Gbps and approximately 4 GB on devices greater than 5 Gbps.

Section 6.1.3 of [ST] states the TOE enforces a maximum size on the audit logs, specified as a percentage of log disk space allocated to each log file using the `log-file-size` CLI setting. The amount of local log disk space allocated to each log cannot exceed the configured percentage and the combined percentage configured for the logs must equal 100%.

Section 6.1.3 of [ST] states the audit records on the TOE are protected by database access control and there are no interfaces to modify or delete individual audit records. Administrators in the Super User role can use the `clear log-file` CLI command to delete locally stored audit log files.

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

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

2.1.3.2 Guidance Activities

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

Section 6.1.3 of [ST] states the audit records on the TOE are protected by database access control and there are no interfaces to modify or delete individual audit records. As such, no configuration is required to protect the locally stored audit data against unauthorized modification or deletion.

2.1.3.3 Test Activities

The evaluator shall perform the following tests:

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

The evaluator verified that the TOE requires the user to be authenticated as Security Administrator to access the audit trail. Unauthenticated access to the audit trail is not allowed by the TOE.

Test 2: The evaluator shall access the audit trail as an authorized administrator and attempt to delete the audit records. The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records authorized for deletion are deleted.

The evaluator accessed the audit trail and viewed the logs then attempted to clear the log. The evaluator queried the logs again to verify that they were cleared.

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

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

2.1.4 Protected Audit Event Storage (FAU_STG_EXT.1)

2.1.4.1 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 6.1.4 of [ST] ("FAU_STG_EXT.1: Protected Audit Event Storage") describes the use of SSH to transmit audit data to an external syslog server. When configured to behave in this manner, the TSF will write audit records externally at the same time they are generated locally.

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 6.1.4 of [ST] states each TOE device (including vTPS) allocates approximately one eighth of its internal disk space for local storage of audit records. Devices that support less than 5 Gbps inspection throughput have 8 GB internal disk space while devices that support 5 Gbps and above inspection throughput have 32 GB internal disk space. This implies approximately 1 GB audit log file disk space on devices with less than 5 Gbps and approximately 4 GB on devices greater than 5 Gbps.

Section 6.1.4 of [ST] states when audit storage space is exhausted, the TOE overwrites previous audit records by deleting the oldest historical log file, renaming the current log file to be a historical file, and creating a new current log file. By default, the TOE maintains five files for log rollover functionality, with each file allocated 20% of the total space allocated for that log.

Section 6.1.3 of [ST] states the audit records on the TOE are protected by database access control and there are no interfaces to modify or delete individual audit records. Administrators in the Super User role can use the `clear log-file` CLI command to delete locally stored audit log files.

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

Section 6.1 of [ST] (“Security Audit”) states the TOE is a standalone device that stores audit records locally. As such, the activities pertaining to distributed TOEs are not applicable.

> 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.

The option ‘overwrite previous audit record’ is selected in [ST]. As mentioned above, the overwrite behavior is described in section 6.1.4 of [ST].

> 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 6.1.4 of [ST] states audit records are transmitted to the external audit server at the same time they are written to the local audit trail.

> 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).

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

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.

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

2.1.4.2 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 "SSH configuration" of [CLI] states the TOE can be configured to send syslog messages over SSH using the "Remote System Log" contact. It describes the commands the administrator uses to configure the TOE to communicate with the external syslog server over SSH and to enable the TOE to send logs to the configured external syslog server. It directs the administrator to consult the applicable documentation to configure cryptographic parameters for an SSH remote syslog server that is not a TippingPoint device.

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 2.5.2 of [CCECG] ("Configuring Log Size/Rotation Settings") states the TOE stores audit records locally and can also be configured to send audit records to an external syslog server using SSH. When configured to send audit records to a syslog server, audit records are written to the external syslog as they are written locally to the device's audit log.

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 5.2.1.5 of [ST] ("Protected Audit Event Storage (FAU_STG_EXT.1)") specifies for FAU_STG_EXT.1.3 that the TOE overwrites previous audit records according to the following rule: the oldest historical audit file is deleted, the current audit file is renamed as a historical audit file, and a new audit file is created.

Section 2.5.2 of [CCECG] states the limits of the Audit and System logs are specified as a percentage of internal log disk space using the `log-file-size` CLI setting. The maximum amount of audit data that are stored locally in each log cannot exceed this percentage and the combined percentage configured for the logs must equal 100%. The log rotation function allows administrators to further control the amount of audit records that are stored. The administrator can specify the maximum size of a log file using the `maxFileSize` parameter and the number of files kept in the log rotation using the `numfiles` parameter.

2.1.4.3 Test Activities

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

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 connect to an audit server (Rsyslog 8.32.0) over SSH. The evaluator then performed actions on the TOE to generate audit records and observed that they were received by the audit server and were protected over SSH. The evaluator also verified the transfer of audit data occurred automatically without further intervention from the evaluator once the external audit server was configured.

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 fill up the local audit trail and verified that when the local audit trail filled the oldest historical file was deleted.

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

[ST] does not claim FAU_STG_EXT.2/LocSpace. Therefore, this test is not applicable to the TOE.

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.

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

2.1.5 Action in Case of Possible Audit Data Loss (FAU_STG_EXT.3/LocSpace)

2.1.5.1 TSS Activities

The evaluator shall examine the TSS to ensure that it details how the Security Administrator is warned before the local storage for audit data is full.

Section 6.1.5 of [ST] ("FAU_STG_EXT.3/LocSpace: Action in Case of Possible Audit Data Loss") states an alert is written to the audit trail when available audit storage exceeds 75% full.

For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how each TOE component realises this SFR. Since this SFR is optional, it might only apply to some TOE components but not all. This might lead to the situation where all TOE components store their audit information themselves but FAU_STG_EXT.3/LocSpace is supported only by one of the components. In particular, the evaluator has to verify, that the TSS describes for every component supporting this functionality, whether the warning is generated by the component itself or through another component and name the corresponding component in the latter case. The evaluator has to verify that the TSS makes clear any situations in which audit records might be 'invisibly lost'.

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

2.1.5.2 Guidance Activities

The evaluator shall also ensure that the guidance documentation describes how the Security Administrator is warned before the local storage for audit data is full and how this warning is displayed or stored (since there is no guarantee that an administrator session is running at the time the warning is issued, it is probably stored in the log files). The description in the guidance documentation shall correspond to the description in the TSS.

Section 2.5.2 of [CCECG] ("Configuring Log Size/Rotation Settings") states the TOE generates an audit record warning that is written to the audit trail when the space allocated for storage of audit records exceeds 75% of capacity. This is not configurable. The administrator can view the audit record by issuing the `show log-file system` CLI command.

2.1.5.3 Test Activities

The evaluator shall verify that a warning is issued by the TOE before the local storage space for audit data is full.

The evaluator performed actions to fill up the audit trail and observed that upon reaching 75% full the TOE generates an audit record as a warning that the local allotted space is almost full.

For distributed TOEs the evaluator shall verify the correct implementation of display warning for local storage space for all TOE components that are supporting this feature according to the description in the TSS. The evaluator shall verify that each component that supports this feature according to the description in the TSS is capable of generating a warning itself or through another component.

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

2.2 Cryptographic Support (FCS)

The following table lists the cryptographic functions supported by the TOE and associated SFRs, the specific algorithms that are claimed for these functions, and the relevant CAVP certificate validation lists and certificate numbers for each.

Functions	Standards	Certificates
FCS_CKM.1 Cryptographic Key Generation		
RSA (2048 bits)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3	A5111: RSA KeyGen (FIPS186-4)
ECC key pair generation (NIST curves P-256, P-384, P-521)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4	A5111: ECDSA KeyGen (FIPS186-4)
FFC schemes using ‘safe-prime’ groups (2048, 3072, 4096 bits)	NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and RFC 3526	CCTL Tested.
FCS_CKM.2 Cryptographic Key Establishment		
ECDSA (P-256, P-384, P-521 curves)	NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”	A5111: KAS
FFC schemes using ‘safe-prime’ groups (2048, 3072, 4096 bits)	NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and RFC 3526	CCTL Tested.
FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)		
AES-CBC (128, 256 bits)	ISO 18033-3, CBC as specified in ISO 10116	A5111: AES-CBC
AES-GCM (128, 256 bits)	ISO 18033-3, GCM as specified in ISO 19772	A5111: AES-GCM

Functions	Standards	Certificates
FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)		
RSA Digital Signature Algorithm (rDSA) (modulus 2048)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5	A5111: RSA SigGen (FIPS 186-4) A5111: RSA SigVer (FIPS 186-4)
ECDSA with NIST curves P-256, P-384, and P-521	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6	A5111: ECDSA SigGen (FIPS 186-4) A5111: ECDSA SigVer (FIPS 186-4)
FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)		
SHA-1 (digest sizes 160 bits) SHA-256 (digest size 256 bits) SHA-384 (digest size 384 bits) SHA-512 (digest size 512 bits)	ISO/IEC 10118-3:2004	A5111: SHA-1 A5111: SHA2-256 A5111: SHA2-384 A5111: SHA2-512
FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)		
HMAC-SHA-1 (key size 160 bits, digest size 160 bits) HMAC-SHA-256 (key size 256 bits, digest size 256 bits) HMAC-SHA-512 (key size 512 bits, digest size 512 bits)	ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”	A5111: HMAC-SHA1-1 A5111: HMAC-SHA2-256 A5111: HMAC-SHA2-512
FCS_RBG_EXT.1 Random Bit Generation		
CTR_DRBG (AES) with two independent platform-based noise source of 256 bits of non-determinism	ISO/IEC 18031:2011	A5111: Counter DRBG

2.2.1 Cryptographic Key Generation (FCS_CKM.1)

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 6.2.1 of [ST] (“FCS_CKM.1: Cryptographic Key Generation”) identifies the key sizes supported by the TOE. The TOE supports the following key generation schemes and their usage:

- RSA schemes using cryptographic key sizes of 2048 bits, for SSH public key authentication.
- ECC schemes using NIST curves P-256, P-384, and P-521, for SSH public key authentication and SSH key establishment.
- FFC schemes using “safe-prime” groups with key sizes of 2048 bits (Diffie-Hellman group 14), 3072 bits (Diffie-Hellman group 15), and 4096 bits (Diffie-Hellman group 16), for SSH key establishment.

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 2.6.2 of [CCECG] ("SSH Configuration") states SSH ciphers are configurable and describes the `debug ssh ciphers` command to enable and disable individual cryptographic algorithms.

The guidance identifies the TOE in its evaluated configuration uses the following public key algorithms for SSH authentication: ssh-rsa; ecdsa-sha2-nistp256; ecdsa-sha2-nistp384; and ecdsa-sha2-nistp521. This is consistent with the ST.

The guidance additionally identifies the TOE in its evaluated configuration uses the following key establishment schemes, consistent with the ST:

- for SSH client—diffie-hellman-group14-sha1; ecdh-sha2-nistp256; ecdh-sha2-nistp384; and ecdh-sha2-nistp521
- for SSH server—diffie-hellman-group14-sha1; diffie-hellman-group15-sha512; and diffie-hellman-group16-sha512.

The guidance states the supported key establishment schemes are not configurable.

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.

Section 6.2 of [ST] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") 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, C.3 Public Exponent Mode: Fixed Fixed Public Exponent: 10001	A #5111 RSA KeyGen (FIPS186-4)
ECC schemes using "NIST curves" P-256 and P-384, that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	Curves: P-256, P-384, P-521	A #5111 ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4)

Modified in accordance with 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 Cryptographic Key Establishment (FCS_CKM.2)

2.2.2.1 TSS Activities

Modified in accordance with TD0580.

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

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 6.2.2 of [ST] (“FCS_CKM.2: Cryptographic Key Establishment”) states the TOE performs key establishment when negotiating an SSH connection using:

- Diffie-Hellman group 14 that implements 2048-bit MODP Group according to RFC 3526, Section 3
- Diffie-Hellman group 15 that implements 3072-bit MODP Group according to RFC 3526, Section 3
- Diffie-Hellman group 16 that implements 4096-bit MODP Group according to RFC 3526, Section 3
- Elliptic Curve Diffie-Hellman key agreement using P-256, P-384, or P-521 curves.

The TOE uses these key establishment methods during SSH session establishment with an external audit server (TOE acts as SSH client) and with users accessing the SSH management interface (TOE acts as SSH server). These key establishment schemes are consistent with the key generation schemes specified in FCS_CKM.1.

2.2.2.2 Guidance Activities

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

Section 2.6.2 of [CCECG] (“SSH Configuration”) states the following SSH key exchange methods are not configurable: diffie-hellman-group14-sha1; diffie-hellman-group15-sha512; diffie-hellman-group16-sha512; ecdh-sha2-nistp256; ecdh-sha2-nistp384; and ecdh-sha2-nistp521.

2.2.2.3 Test Activities

Modified in accordance with 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] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") 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 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	Scheme: Ephemeral Unified: KAS Role: Initiator, Responder Parameter Sets: EC: Curve: P-256 SHA: SHA2-256 ED: Curve: P-384 SHA: SHA2-384 EE: Curve: P-521 SHA: SHA2-512	A #5111 KAS

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.

N/A – the TOE does not implement RSA-based key establishment schemes.

Diffie-Hellman Group 14

The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses Diffie-Hellman group 14.

The TOE uses Diffie-Hellman group 14 for SSH client and server functionality, the functionality of this key exchange for the SSH protocol has been verified by the evaluation team using the known good implementation of OpenSSH 7.6p1 to establish connections with the TOE.

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.

The TOE uses the following safe-prime groups for Diffie Hellman of group 14, 16, 18 for SSH server functionality and group 14 for SSH client functionality. The functionality of this key exchange for the SSH protocol has been verified by the evaluation team using the known good implementation of OpenSSH 7.6p1 to establish connections with the TOE.

2.2.3 Cryptographic Key Destruction (FCS_CKM.4)

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 6.2.3 of [ST] ("FCS_CKM.4: Cryptographic Key Destruction"), Table 6 ("Secret keys, Private keys and CSPs") lists the secret keys and private keys used by the TOE. Section 6.2.3 states user passwords and SSH RSA client keys are stored in internal flash, encrypted using AES with a 256 bit Key Encrypting Key (KEK). The KEK is stored on Compact Flash and is itself encrypted using AES with a 256 bit Master Key. All other keys are plaintext stored in volatile memory and destroyed automatically through a single overwrite of zeroes. For TPS appliances, the Master Key exists in hardware circuitry within the TOE. It is generated during manufacturing and is unique to each appliance. For the vTPS, the Mater Key is generated during software installation and stored on a system memory file. The vTPS implements multi-layer software obfuscation techniques (including masking and key wrapping) to protect the Master Key. These techniques protect the Master Key and associated authorization factors from unauthorized access. Anyone who has the copy of software or access to a running copy of software will not be able to access the plaintext Master Key by visually inspecting the software image, reverse engineering the software, or inspecting a memory footprint of a running software image. The Master Key and associated authorization factors are destroyed when the vTPS is factory reset (by execution of the `debug factory-reset` CLI command, or by deleting and reinstalling the VM). When this occurs, a new Master Key and authorization factors are generated, overwriting the previous values.

The evaluators reviewed the TSS sections relating to protection of TSF data and cryptographic communications and did not observe any behavior that would suggest keys are absent from this discussion.

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).

All keys identified by the ST are plaintext keys stored in volatile memory or encrypted keys stored in non-volatile memory. There are no plaintext keys stored by the TOE in non-volatile memory.

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.

All keys identified by the ST are plaintext keys stored in volatile memory or encrypted keys stored in non-volatile memory. There are no plaintext keys stored by the TOE in non-volatile memory.

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 6.2.3 of [ST] identifies user passwords and SSH RSA client keys as being stored in a non-plaintext form. The TSS identifies the encryption method as AES with a 256-bit symmetric key as the key-encrypting-key (KEK). The KEK is itself stored in an encrypted form on Compact Flash (CF) using AES. The key used to encrypt the KEK exists in hardware circuitry within the TOE.

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 TSS does not identify any circumstances that do not conform to the key destruction requirement.

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.

The ST does not specify the use of "a value that does not contain any CSP" to overwrite keys. This activity is therefore not applicable.

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

This is N/A because the TSF does not have any circumstances where the key destruction requirement is not met as claimed.

2.2.3.3 Test Activities

None defined.

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

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.

Section 6.2.4 of [ST] ("FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/ Decryption)") identifies the key sizes the TOE uses for data encryption and decryption as 128 and 256 bits, and the modes as CBC and GCM.

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 2.4 of [CCECG] ("Configuring the TPS for Common Criteria Compliance") states the TOE must be configured to support FIPS 140-2 cryptographic requirements. The TOE provides the `fips-mode-enable` CLI command to enable FIPS mode on the TOE. FIPS mode restricts cryptographic mechanisms to FIPS-approved algorithms.

Section 2.6.2 of [CCECG] ("SSH Configuration") states SSH ciphers are configurable and provides the `debug ssh ciphers` command to enable and disable individual cryptographic algorithms. The administrator uses this command to enable or disable the *aes128-cbc*, *aes256-cbc*, *aes128-gcm@openssh.com*, and *aes256-gcm@openssh.com* encryption algorithms for use in SSH.

2.2.4.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") identifies the CAVP certifications verifying AES encryption and decryption, as follows.

Algorithm	Tested Capabilities	Certificates
AES-CBC as defined in ISO 10116	Direction: Decrypt, Encrypt Key Length: 128, 256	A #5111 AES-CBC
AES-GCM as defined in ISO 19772	Direction: Decrypt, Encrypt IV Generation: Internal Key Length: 128, 256	A #5111 AES-GCM

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

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.

Section 6.2.5 of [ST] ("FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)") states the TOE provides cryptographic signature services using RSA Digital Signature Algorithm with key size of 2048 bits and uses ECDSA implementing NIST curves P-256, P-384, and P-521 in support of SSH public key authentication.

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 2.4 of [CCECG] ("Configuring the TPS for Common Criteria Compliance") states the TOE must be configured to support FIPS 140-2 cryptographic requirements. The TOE provides the `fips-mode-enable` CLI command to enable FIPS mode on the TOE. FIPS mode restricts cryptographic mechanisms to FIPS-approved algorithms.

2.2.5.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") identifies the CAVP certifications verifying digital signature generation and verification, as follows.

Algorithm	Tested Capabilities	Certificates
RSA Digital Signature Algorithm with 2048 bit modulus as defined in FIPS PUB 186-4	**RSA Signature Generation** Signature Type: PKCS 1.5 Modulo: 2048 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Signature Type: PKCSPSS Modulo: 2048	A #5111

	Hash: SHA2-256; Salt Length: 0 Hash: SHA2-384; Salt Length: 0 Hash: SHA2-512; Salt Length: 0	
ECDSA using NIST curves P-256, P-384, and P-521 as defined in FIPS PUB 186-4		A #5111

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

2.2.6.1 TSS Activities

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

Section 6.2.6 of [ST] ("FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)") states the TOE performs SHA-1, SHA-256, SHA-384, and SHA-512 cryptographic hashing services as part of HMAC and RSA and ECDSA digital signature generation and verification.

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 2.4 of [CCECG] ("Configuring the TPS for Common Criteria Compliance") states the TOE must be configured to support FIPS 140-2 cryptographic requirements. The TOE provides the `fips-mode-enable` CLI command to enable FIPS mode on the TOE. FIPS mode restricts cryptographic mechanisms to FIPS-approved algorithms.

2.2.6.3 Test Activities

> Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") identifies the CAVP certifications verifying cryptographic hashing, as follows.

Algorithm	**Tested Capabilities**	**Certificates**
SHS as defined in ISO/IEC 10118-3:2004	SHA-1 SHA-256 SHA-384 SHA-512	A #5111 SHA-1 SHA-256 SHA-384 SHA-512

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

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.

Section 6.2.7 of [ST] (“FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)”) states the HMAC function implemented by the TOE uses key lengths, hash function, block size, and output MAC length as summarized in the following table:

Algorithm	Key Size	Block Size	Message Digest Size
SHA-1	160	512	160
SHA-256	256	512	256
SHA-512	512	1024	512

This section also states that implicit keyed hash message authentication is used when the TOE’s SSH implementation uses AES-GCM for encryption.

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 2.4 of [CCECG] (“Configuring the TPS for Common Criteria Compliance”) states the TOE must be configured to support FIPS 140-2 cryptographic requirements. The TOE provides the `fips-mode-enable` CLI command to enable FIPS mode on the TOE. FIPS mode restricts cryptographic mechanisms to FIPS-approved algorithms.

2.2.7.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] (“Cryptographic Support”), Table 5 (“Cryptographic Functions”) identifies the CAVP certifications verifying cryptographic keyed hashing, as follows.

Algorithm	Tested Capabilities	Certificates
HMAC that meets ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”	HMAC-SHA1 Key sizes < block size Key sizes > block size Key size = block size HMAC-SHA2-256 Key sizes < block size Key sizes > block size Key size = block size HMAC-SHA2-512 Key sizes < block size Key sizes > block size Key size = block size	A #5111 HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-512

2.2.8 Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)

2.2.8.1 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 6.2.8 of [ST] ("FCS_RBG_EXT.1: Random Bit Generation") states the TOE uses an AES counter DRBG for random bit generation services. It further states the TOE seeds the DRBG with 256 bits of entropy and all platforms use entropy provided by the Linux kernel, including device, input, interrupt, disk randomness, and the RDRAND instruction.

2.2.8.2 Guidance Activities

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

Section 2.4 of [CCECG] ("Configuring the TPS for Common Criteria Compliance") states the TOE is required to be configured into FIPS mode. FIPS-CC Mode restricts the cryptographic mechanisms to FIPS-approved algorithms. No further configuration is required to ensure the use of FIPS approved algorithms or to configure RNG functionality.

2.2.8.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") identifies the CAVP certifications 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	A #5111 Counter DRBG

2.2.9 SSH Client (FCS_SSHC_EXT.1)

2.2.9.1 TSS Activities

Modified in accordance with TD0636.

FCS_SSHC_EXT.1.2

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

Section 6.2.9 of [ST] ("FCS_SSHC_EXT.1 – SSH Client Protocol / FCS_SSHS_EXT.1 – SSH Server Protocol") states the TOE's SSH client implementation supports RSA for its SSH local user key (used for user authentication to an external SSH server). This list is consistent with:

- The asymmetric key generation algorithm selected in FCS_CKM.1 (RSA).
- The hashing algorithms selected in FCS_COP.1/Hash (SHA-256, SHA-384, SHA-512)
- The signature generation algorithms selected in FCS_COP.1/SigGen (RSA).

Section 6.2.9 of [ST] states the TOE acts as an SSH client for secure communications with an external audit server and that the TOE's SSH client implementation supports the public key-based authentication method.

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

The ST does not select password-based authentication methods in FCS_SSHC_EXT.1.2.

FCS_SSHC_EXT.1.3

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

Section 6.2.9 of [ST] states the TOE drops packets larger than 256K bytes in an SSH transport connection. As it receives SSH packets, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. If the packet is incomplete when the buffer becomes full (256K bytes), the packet is dropped.

FCS_SSHC_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 6.2.9 [ST] states the TOE's SSH transport implementation uses the following encryption algorithms: aes128-cbc; aes256-cbc; aes128-gcm@openssh.com; and aes256-gcm@openssh.com. This list is identical to the algorithms specified in the functional requirement.

No optional characteristics for SSH are specified in [ST].

Modified in accordance with TD0636.

FCS_SSHC_EXT.1.5

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

Section 6.2.9 of [ST] states the administrator uses the `ssh-host-key` command during configuration of the TOE's SSH client connection to a remote SSH server to specify the host public key associated with that connection.

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

Section 6.2.9 of [ST] states the TOE's SSH transport implementation uses the following public key algorithms for authentication: ssh-rsa; ecdsa-sha2-nistp256; ecdsa-sha2-nistp384; and ecdsa-sha2-nistp521. This list conforms to the algorithms selected in FCS_SSHC_EXT.1.5.

No optional characteristics for SSH are specified in [ST].

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

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

FCS_SSHC_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 6.2.9 of [ST] states the TOE's SSH transport implementation uses the following data integrity algorithms: hmac-sha1; hmac-sha2-256; hmac-sha2-512; implicit (when aes*-gcm@openssh.com is used as the public key algorithm). This list conforms to the algorithms selected in FCS_SSHC_EXT.1.6.

FCS_SSHC_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 6.2.9 of [ST] states the TOE's SSH client uses the following key exchange algorithms: diffie-hellman-group14-sha1; ecdh-sha2-nistp256; ecdh-sha2-nistp384; and ecdh-sha2-nistp521. This list conforms to the algorithms selected in FCS_SSHC_EXT.1.7.

FCS_SSHC_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 6.2.9 of [ST] states the TOE ensures the SSH connection is rekeyed either when a threshold of one hour has been reached, or when one gigabyte of data has been transmitted. Both thresholds are checked by the TOE and rekeying is performed upon reaching whichever threshold is hit first.

2.2.9.2 Guidance Activities

Added in accordance with TD0636.

FCS_SSHC_EXT.1.2

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

Section 2.6.2 of [CCECG] ("SSH Configuration") contains instructions to the administrator to configure SSH to ensure only the allowed mechanisms are used in SSH connections initiated by the TOE.

FCS_SSHC_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 2.6.2 of [CCECG] states the SSH ciphers can be enabled and disabled via the `debug ssh ciphers` CLI command.

FCS_SSHC_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).

Section 2.6.2 of [CCECG] lists the public key algorithms the SSH client implementation supports. Section 2.6.2.1 of [CCECG] ("SSH Host Key Configuration") describes how the administrator configures the trusted public keys database for a remote SSH server.

FCS_SSHC_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 2.6.2 of [CCECG] lists the key exchange methods supported by the TOE in FIPS mode. The TOE does not require any configuration either to specify or to restrict the TOE to using only the listed algorithms. It states the "none" MAC algorithm is not allowed and there are no other configuration options.

FCS_SSHC_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.

Section 2.6.2 of [CCECG] lists the key exchange methods supported by the TOE in FIPS mode and states they are not configurable.

FCS_SSHC_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 6.2.9 of [ST] indicates the SSH rekeying thresholds are not configurable. The SSH connection is rekeyed either when a threshold of one hour has been reached, or when one gigabyte of data has been transmitted. Both thresholds are checked by the TOE and rekeying is performed upon reaching whichever threshold is hit first.

2.2.9.3 Test Activities

Modified in accordance with TD0636.

FCS_SSHC_EXT.1.2

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

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

The only claimed user authentication method is public key-based and RSA is the only supported public key algorithm. The evaluator verified that the TOE could present and use an RSA public key to successfully authenticate itself to a remote SSH server.

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

Password-based authentication methods have not been selected in the ST. Therefore, this test activity is not applicable to the TOE.

FCS_SSHC_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 established a connection from the TOE to an SSH server configured to be able to send large packets. Once the SSH connection was established, the evaluator sent a packet larger than 256K bytes from the server to the TOE. The evaluator observed that the TOE did not act upon the large packet. Instead, the TOE disconnected the session after the packet was fully sent, and initiated a new connection to the server.

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

The evaluator established a connection from the TOE to an SSH server and used a packet capture tool to collect the network traffic exchanged between the TOE and the SSH server. The evaluator confirmed the TOE offered to the server all the algorithms and only the algorithms specified in the TSS. The evaluator observed successful negotiation and establishment of an SSH session between the TOE and the external SSH server.

FCS_SSHC_EXT.1.5

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

The evaluator established a connection from the TOE to an SSH server using each of the public key algorithms specified in the requirement. The evaluator confirmed via packet capture successful negotiation of the algorithm and the successful authentication of the TOE by the SSH server.

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

The evaluator configured the SSH server to accept only a public key algorithm not supported by the TOE. The evaluator confirmed via packet capture that attempts to establish an SSH connection from the TOE to the SSH server were rejected.

FCS_SSHC_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 established a connection from the TOE to an SSH server using each of the integrity algorithms specified in the SFR. The evaluator confirmed via packet capture successful negotiation of each algorithm and the successful establishment of the connection from the TOE to the SSH server.

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

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 configured the SSH server to accept only a MAC algorithm not supported by the TOE. The evaluator confirmed via packet capture that attempts to establish an SSH connection from the TOE to the SSH server failed.

FCS_SSHC_EXT.1.7

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

The evaluator established a connection from the TOE to an SSH server using each of the key exchange methods specified in the SFR. The evaluator confirmed via packet capture successful negotiation of each algorithm and the successful establishment of the connection from the TOE to the SSH server.

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

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

For testing of the traffic-based threshold the evaluator shall use the TOE to connect to an SSH server, and shall transmit data from and to the TOE within the active SSH session until the threshold for transmitted traffic is reached. The transmitted traffic is the total traffic comprising incoming and outgoing traffic.

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 server the TOE is connected to.

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 to connect to a proprietary SSH server, which does not implement the rekeying initiation logic. The evaluator observed that the TOE initiated a rekey at either 1 hour of the session being open or 1 Gigabyte of data being transferred across the session.

Rekey thresholds are not configurable, so the portion of the testing related to this is not applicable.

FCS_SSHC_EXT.1.9

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

The evaluator deleted all entries in the TOE’s list of recognized SSH server host keys. The evaluator then initiated an SSH connection from the TOE to the SSH server. The evaluator confirmed via packet captures and log records that the TOE rejected the connection and did not establish a SSH connection.

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

The evaluator configured the TOE with the peer SSH server’s host key. The evaluator then altered the SSH server’s host key. The evaluator initiated an SSH connection from the TOE to the external SSH server. The evaluator confirmed the TOE rejected the connection.

2.2.10 SSH Server (FCS_SSHS_EXT.1)

2.2.10.1 TSS Activities

Modified in accordance with TD0631.

FCS_SSHS_EXT.1.2

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).

Section 6.2.9 of [ST] (“FCS_SSHC_EXT.1 – SSH Client Protocol / FCS_SSHS_EXT.1 – SSH Server Protocol”) states the TOE’s SSH server implementation accepts the following public key algorithms for client authentication: ssh-rsa; ecdsa-sha2-nistp256; ecdsa-sha2-nistp384; and ecdsa-sha2-nistp521. This list is consistent with the signature verification algorithms selected in FCS_COP.1/SigGen (RSA, ECDSA).

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.

Section 6.2.9 of [ST] states the administrator uses the `ssh-public-key` command within the `aaa user` context to associate a public key with a user account on the TOE.

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 6.2.9 of [ST] states the TOE's SSH server implementation supports password-based authentication as described in RFC 4253.

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 6.2.9 of [ST] states the TOE drops packets larger than 256K bytes in an SSH transport connection. As it receives SSH packets, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. If the packet is incomplete when the buffer becomes full (256K bytes), the packet is dropped.

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 6.2.9 of [ST] states the TOE's SSH transport implementation uses the following encryption algorithms: aes128-cbc; aes256-cbc; aes128-gcm@openssh.com; aes256-gcm@openssh.com. This list is identical to the algorithms specified in the functional requirement.

No optional characteristics for SSH are specified in [ST].

Modified in accordance with TD0631.

FCS_SSHS_EXT.1.5

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 6.2.9 of [ST] states the TOE's SSH transport implementation for its SSH server implementation uses ssh-rsa, rsa-sha2-256, and rsa-sha2-512 as its public key algorithms for authentication. This list conforms to the algorithms selected in FCS_SSHS_EXT.1.5.

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 6.2.9 of [ST] states the TOE's SSH transport implementation for its SSH server implementation uses the following data integrity algorithms: hmac-sha2-256; hmac-sha2-512; implicit (when aes*-

gcm@openssh.com is used as the encryption algorithm). This list conforms to the algorithms selected in 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 6.2.9 of [ST] states the TOE uses the following key exchange algorithms: ecdh-sha2-nistp256; diffie-hellman-group14-sha256; diffie-hellman-group16-sha512; diife-hellman-group18-sha512; ecdh-sha2-nistp384; and ecdh-sha2-nistp521. This list conforms to the algorithms selected in FCS_SSHS_EXT.1.7.

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 6.2.9 of [ST] states the TOE ensures the SSH connection is rekeyed either when a threshold of one hour has been reached, or when one gigabyte of data has been transmitted. Both thresholds are checked by the TOE and rekeying is performed upon reaching whichever threshold is hit first.

2.2.10.2 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 2.6.2 of [CCECG] ("SSH Configuration") states the SSH encryption algorithms can be enabled and disabled via the `debug ssh ciphers` command.

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).

Section 2.6.2 of [CCECG] states ssh-rsa, rsa-sha2-256, and rsa-sha2-512 are the public key algorithms supported by the TOE's SSH server implementation.

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 2.6.2 of [CCECG] lists the key exchange methods supported by the TOE in FIPS mode. The TOE does not require any configuration either to specify or to restrict the TOE to using only the listed algorithms. It states the "none" MAC algorithm is not allowed and there are no other configuration options.

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.

Section 2.6.2 of [CCEGC] lists the key exchange methods supported by the TOE in FIPS mode and that they are not configurable.

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 6.2.9 of [ST] indicates the SSH rekeying thresholds are not configurable. The SSH connection is rekeyed either when a threshold of one hour has been reached, or when one gigabyte of data has been transmitted. Both thresholds are checked by the TOE and rekeying is performed upon reaching whichever threshold is hit first.

2.2.10.3 Test Activities

Modified in accordance with TD0631.

FCS_SSHS_EXT.1.2

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

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 evaluator configured a public key-based authentication method for a user. The evaluator verified that the TOE granted the user access over SSH when the correct public key-based authentication was attempted by the SSH client. The evaluator performed this test for each supported client public-key authentication algorithm (ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521).

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 evaluator generated a new keypair and did not configure the new key on the TOE. The evaluator verified that the TOE did not grant the user access over SSH when the unknown public key-based authentication was attempted by the SSH client.

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 verified that the TOE grants access to users over SSH when the correct username/password combination is presented in the SSH channel.

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 verified that the TOE does not grant access to users over SSH when the incorrect username/password combination is presented in the SSH channel.

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 established a connection to the TOE from an SSH client configured to be able to send large packets. Once the SSH connection was established, the evaluator sent a packet larger than 256K bytes from the client to the TOE. The evaluator observed that the TOE did not act upon the large packet, and instead disconnected the session.

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 established a connection to the TOE from an SSH client and used a packet capture tool to collect the network traffic exchanged between the SSH client and the TOE. The evaluator confirmed the TOE offered to the client all the algorithms and only the algorithms specified in the TSS. The evaluator observed successful negotiation and establishment of an SSH session between the SSH client and the TOE.

Modified in accordance with TD0631.

FCS_SSHS_EXT.1.5

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 evaluator verified that the TOE could use each of the claimed algorithms to authenticate itself to the SSH client.

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 verified that a connection to the TOE could not be established when the client did not offer to use a Host key algorithm supported by the TOE.

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 established a connection to the TOE from an SSH client using each of the integrity algorithms specified in the SFR. The evaluator confirmed via packet capture successful negotiation of each algorithm and the successful establishment of the connection from the SSH client to the TOE.

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 configured an SSH client to attempt to connect to the TOE using only a MAC algorithm which is not specified in the requirement. The TOE denied the connection attempt.

FCS_SSHS_EXT.1.7

Test 1: The evaluator shall configure an SSH client to only allow the diffiehellman-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 configured an SSH client to attempt to connect to the TOE using only the diffie-hellman-group1-sha1 key exchange. The TOE denied the connection attempt.

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 established a connection to the TOE from an SSH client using each of the key exchange methods specified in the SFR. The evaluator confirmed via packet capture the successful establishment of the connection from the SSH client to the TOE.

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 an SSH client to connect to the TOE, and shall transmit data from and to the TOE within the active SSH session until the threshold for transmitted traffic is reached. The transmitted traffic is the total traffic comprising incoming and outgoing traffic.

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 connected to the TOE with a proprietary SSH client which does not implement rekey initiation attempts. The evaluator verified that the TOE initiated a rekey of a SSH session at 1 hour of a session being open or after 1 Gigabyte of data being sent in the session.

Rekey thresholds are not configurable, so the portion of the testing related to this is not applicable.

2.3 Identification and Authentication (FIA)

2.3.1 Authentication Failure Management (FIA_AFL.1)

2.3.1.1 TSS Activities

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

The TOE supports one method of remote administrative access to the TOE, using SSH.

Section 6.3.1 of [ST] (“FIA_AFL.1 Authentication Failure Management”) states the TOE detects when an administrator-configurable number (from 1 to 10) of failed remote authentication attempts has been reached. When the configured number of unsuccessful authentication attempts has been reached, the affected administrator account is locked for an administrator-configurable period of time (in the range 1 to 1,440 minutes).

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 6.3.1 of [ST] states authentication failures by remote administrators cannot lead to a situation where no administrator access to the TOE is available. The authentication failure management function applies only to remote attempts to access the TOE. The TOE is always accessible to an authenticated administrator via the local console.

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 “Command Line Interface > Edit running configuration commands > Contexts and related commands > running-aaa Context Commands > ips{running-aaa}login” of [CLI] provides instructions for configuring the number of successive unsuccessful authentication attempts and the period of time (in minutes) the affected account is locked.

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 2.6.5 of [CCECG] (“Authentication Failure Handling”) states authentication failures by remote Administrators cannot lead to a situation where no Administrator access to the TOE is available. If remote administrators are locked out, administrator access is still available via the local console.

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 only form of remote administration supported by the TOE is accessing the CLI via SSH.

The evaluator configured the TOE to lockout a user after a specified number of failed login attempts. The evaluator made sufficient attempts to authenticate remotely to the CLI using invalid credentials, such that the user would be locked out. The evaluator verified that the user was unable to authenticate to the TOE with valid credentials once the configured invalid authentication attempts value had been reached.

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 ST only includes the time period selection. The evaluator attempted to login to the TOE with valid credentials before the time period elapsed and confirmed the TOE did not allow access. The evaluator then waited for the configured number of minutes to elapse and then attempted to remotely login to the CLI. The evaluator verified that the user was able to authenticate once again to the TOE.

2.3.2 Password Management (FIA_PMG_EXT.1)

2.3.2.1 TSS Evaluation Activity

Modified in accordance with TD0792.

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

Section 6.3.2 of [ST] ("FIA_PMG_EXT.1: Password Management") lists the supported special characters for the composition of administrator passwords, consistent with the list claimed in FIA_PMG_EXT.1.1.

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

Section 6.3.2 of [ST] states that minimum password length is administrator configurable.

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

Section 6.3.2 of [ST] states that minimum password length is configurable to 1, 8, or 15 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 2.6.4 of [CCECG] ("Password Considerations") identifies the characters permitted in a password, provides suggestions and guidance to the administrator on the composition of strong passwords, provides instructions for configuring the minimum password length, and identifies the valid minimum password lengths the TOE supports.

2.3.2.3 Test Activities

The evaluator shall perform the following tests.

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

The evaluator verified that passwords could be configured on the TOE for a user using each of the claimed characters and symbols. The evaluator also verified that the length requirements of the password are enforced by the TOE.

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

The evaluator verified that the TOE rejected passwords and prevented passwords from being set if the password does not meet the minimum length requirements.

2.3.3 Protected Authentication Feedback (FIA_UAU.7)

2.3.3.1 TSS Evaluation Activity

None defined.

2.3.3.2 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 6.3.3 of [ST] ("FIA_UAU.7: Protected Authentication Feedback") states when an administrator logs in, the TOE does not echo the password as it is entered. There are no preparatory steps required to ensure authentication data is not revealed while entering login information.

2.3.3.3 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.

The evaluator verified that when attempting to authenticate to the TOE at the local console, the TOE does not provide any indication of the password that was used to the user. The TOE displayed only the username that was entered and the result of the authentication attempt.

2.3.4 Password-based Authentication Mechanism (FIA_UAU_EXT.2)

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

2.3.5 User Identification and Authentication (FIA_UIA_EXT.1)

2.3.5.1 TSS Evaluation Activity

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 6.3.4 of [ST] ("FIA_UIA_EXT.1: User Identification and Authentication, FIA_UAU_EXT.2: Password-based Authentication Mechanism") identifies the following methods by which administrators access and manage the TOE: locally, using a directly connected console to access the CLI; locally, using a direct connection to the Ethernet Management port to access the CLI; remotely, via an SSH connection to the Ethernet Management port over a network to access the CLI.

In order to log in to the CLI, either locally or remotely, the administrator provides an identity and authentication data that matches the claimed identity. Users are defined locally within the TOE with a user identity, authentication data (password or public key) and role, and are authenticated by 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 6.3.4 of [ST] states the TOE allows the following actions prior to user identification and authentication: for both local users at the console and remote users over SSH, the TOE displays the configured access banner; the TOE responds to ICMP requests received on the Ethernet Management interface.

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.

The TOE is not distributed so this activity does not apply.

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.

The TOE is not distributed so this activity does not apply.

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.

Section 2.4 of [CCECG] (“Configuring the TPS for Common Criteria Compliance”) describes how initial authentication is accomplished using SSH username and password. Section “Command Line Interface” of [CLI] states that SSH is enabled by default, so no additional configuration is needed to support this behavior. However, section 2.4 of [CCECG] also states that public key authentication can be used for SSH instead of the default username/password method. It then provides a summary on how to configure this behavior and references the relevant section of [CLI], which goes on to provide the instructions in full detail.

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.

The evaluator attempted to log onto the TOE with the incorrect credentials and verified that this attempt was denied. The evaluator then attempted to log onto the TOE with the correct credentials and verified that this attempt succeeded.

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 ran a NMAP scan against the TOE and verified that the only open port was 22 which is used for SSH remote login. The evaluator also confirmed that the TOE displays that logon banner during log in and that the TOE will respond to ICMP request messages.

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 evaluator confirmed that the only service available prior to logon for users requesting local access is the display of the login banner.

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.

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

2.4 Security Management (FMT)

2.4.1 General requirements for distributed TOEs

2.4.1.1 TSS Activities

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 activity does not apply.

2.4.1.2 Guidance Activities

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 activity does not apply.

2.4.1.3 Tests Activities

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 activity does not apply.

2.4.2 Management of Security Functions Behavior (FMT_MOF.1/Functions)

2.4.2.1 TSS Activities

For distributed TOEs see chapter 2.4.1.1 [of [CPP_ND_V2.2-SD]].

The TOE is not distributed so this activity does not apply.

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

Section 5.2.4.2 of [ST] ("Management of Security Functions Behaviour (FMT_MOF.1/Functions)") specifies in FMT_MOF.1/Functions the ability to determine and modify the behavior of the function of transmitting audit data to an external IT entity is restricted to Security Administrators.

Section 6.1.4 of [ST] ("FAU_STG_EXT.1: Protected Audit Event Storage") states the TOE can be configured to transmit audit records to a remote syslog server over SSH. Section 6.4.5 of [ST] ("FMT_SMR.2: Restrictions on Security Roles") states the TOE supports the pre-defined administrator roles Super User, Admin, and Operator, which map to the Security Administrator role defined in [CPP_ND_V2.2E]. All three roles can determine the behavior of the function to transmit audit data to an external IT entity, while the Super User and Admin roles have the ability to modify the behavior of the function to transmit audit data to an external IT entity. Section 6.4.2 of [ST] ("FMT_MOF.1/Functions: Management of Security Functions Behaviour") further states users with the Super User or Admin role can configure the audit data to be transmitted to a remote syslog server.

2.4.2.2 Guidance Activities

For distributed TOEs see chapter 2.4.1.2.

The TOE is not distributed so this activity does not apply.

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

Section "SSH configuration" of [CLI] identifies and describes the commands the Security Administrator uses to configure the TOE to transmit audit records to an external syslog server over SSH, while section "Edit running configuration commands > Edit context commands > display" describes how the administrator can determine the current behavior of the function by displaying the log-configuration context.

2.4.2.3 Test Activities

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

The evaluator confirmed that a user without security administrator permissions, an unauthenticated user, is unable to modify the settings for transmission of audit data to an external IT entity.

Test 2 (if 'transmission of audit data to external IT entity' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity with prior authentication as Security Administrator. The effects of the modifications should be confirmed.

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

The evaluator confirmed that a user with security administrator permissions was able to modify the settings for transmission of audit data to an external IT entity.

Test 1 (if 'handling of audit data' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify all security related parameters for configuration of the handling of audit data without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator. The term 'handling of audit data' refers to the different options for selection and assignments in SFR s FAU_STG_EXT.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace.

This ST does not make this selection, so these test activities are not applicable.

Test 2 (if 'handling of audit data' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify all security related parameters for configuration of the handling of audit data with prior authentication as Security Administrator. The effects of the modifications should be confirmed. The term 'handling of audit data' refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace.

The evaluator does not necessarily have to test all possible values of the security related parameters for configuration of the handling of audit data but at least one allowed value per parameter.

This ST does not make this selection, so these test activities are not applicable.

Test 1 (if 'audit functionality when Local Audit Storage Space is full' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify the behaviour when Local Audit Storage Space is full without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). This attempt should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

This ST does not make this selection, so these test activities are not applicable.

Test 2 (if 'audit functionality when Local Audit Storage Space is full' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify the behaviour when Local Audit Storage Space is full with prior authentication as Security Administrator. This attempt should be successful. The effect of the change shall be verified.

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

This ST does not make this selection, so these test activities are not applicable.

Test 3 (if in the first selection 'determine the behaviour of' has been chosen together with for any of the options in the second selection): The evaluator shall try to determine the behaviour of all options chosen from the second selection without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). This can be done in one test or in separate tests. The attempt(s) to determine the behaviour of the selected functions without administrator authentication shall fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

The evaluator verified that a non-authenticated user is not able to query the TOE to determine the behavior of its functions.

Test 4 (if in the first selection 'determine the behaviour of' has been chosen together with for any of the options in the second selection): The evaluator shall try to determine the behaviour of all options chosen from the second selection with prior authentication as Security Administrator. This can be done in one test or in separate tests. The attempt(s) to determine the behaviour of the selected functions with Security Administrator authentication shall be successful.

The evaluator logged onto the TOE as a user with security administrator permissions and verified that the user was able to determine the behavior of the TOE's functions.

2.4.3 Management of Security Functions Behavior (FMT_MOF.1/ManualUpdate)

2.4.3.1 TSS Activities

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

The TOE is not distributed so this activity does not apply.

2.4.3.2 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 2.9 of [CCECG] ("TOE Updates") describes how to manually initiate an update using the `debug upgrade URL` CLI command. The TOE verifies the update by verifying the signature and hash. If the verification fails, the update will not be installed and the TOE will log an error.

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 activity is not applicable.

2.4.3.3 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 perform an update to the TOE as a user without security administrator permissions, an unauthenticated user, and verified that this attempt was denied.

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

The evaluator verified that the TOE permits users which are Security Administrators to update the TOE with a legitimate image.

2.4.4 Management of TSF Data (FMT_MTD.1/CoreData)

2.4.4.1 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 2.4 of [CCECG] ("Configuring the TPS for Common Criteria Compliance") identifies only banner display and ICMP as the functions that are accessible prior to administrator log-in. Section 6.3.4 of [ST] ("FIA_UIA_EXT.1: User Identification and Authentication, FIA_UAU_EXT.2: Password-based Authentication Mechanism") states the only TSF-mediated actions available prior to logging in are display of the access banner and response to ICMP requests to confirm connectivity. Section 6.4.3 of [ST] ("FMT_MTD.1/CoreData: Management of TSF Data") states no administrative functions are accessible prior to administrator log-in, and only administrators are able to manage TSF data.

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.

The TOE does not claim any X.509v3 certificate functionality so this activity is not applicable.

2.4.4.2 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 functions that manipulate TSF data are listed above in section 2.1.1.2, together with references to the guidance documentation where those functions are identified and described. Section 2.8.1 of [CCECG] ("Administrator Accounts and Roles") states the TOE provides a predefined set of user groups that each have an assigned role with fixed access privileges. The permissions assigned to the default roles/groups cannot be modified. Table 5 of [CCECG] ("Administrator Actions and Role Needed") lists each administrative action mapped to the role needed to perform the action.

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 does not claim any X.509v3 certificate functionality so this activity is not applicable.

2.4.4.3 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.

2.4.5 Specification of Management Functions (FMT_SMF.1)

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.

2.4.5.1 TSS Activities (also including activities for 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 6.4.4 of [ST] ("FMT_SMF.1: Specification of Management Functions") states all administrative functionality is available both locally and remotely via the CLI. The following security-relevant functions are made available at the CLI (along with a number of functions that are outside the scope of the evaluation and therefore not discussed in the TSS):

- Configure the access banner

- Configure the cryptographic functionality (cryptographic ciphers used in SSH sessions)
- Set the time which is used for time-stamps
- Update the TOE, and verify the updates using the digital signature capability prior to installing those updates
- Configure the authentication failure parameters for FIA_AFL.1
- Configure the session inactivity time before session termination
- Configure audit behavior (send audit records to a remote syslog server)
- Manage the trusted public keys database.

> 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 6.3.4 of [ST] ("FIA_UIA_EXT.1: User Identification and Authentication, FIA_UAU_EXT.2: Password-based Authentication Mechanism") states the TOE is administered locally through direct connection to the console interface or Ethernet Management Port. The logical interface to administer the TOE is the CLI, which is the same interface that is accessed remotely via SSH.

Section 2.4 of [CCECG] ("Configuring the TPS for Common Criteria Compliance") states administrators manage the TOE remotely using an SSH connection to the Ethernet Management port on the TOE appliance or locally through the console interface or locally through a direct connection to the Ethernet Management port. Each method provides access to the CLI after an administrator successfully logs in.

> 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 activity is not applicable.

2.4.5.2 Guidance Activities

> See section 2.4.4.1. (2.4.5.1 in this AAR)

This activity was completed in section 2.4.5.1 above.

2.4.5.3 Test Activities

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

The management functions specified in FMT_SMF.1 have been tested as follows:

- Ability to administer the TOE locally and remotely
  - Tested as part of the testing for FMT_SMR.2
- Ability to configure the access banner
  - Tested as part of the testing for FTA_TAB.1
- Ability to configure the session inactivity time before session termination or locking
  - Tested as part of the testing for FTA_SSL_EXT.1 and FTA_SSL.3
- Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;
  - Tested as part of the testing for FPT_TUD_EXT.1

- Ability to configure the authentication failure parameters for FIA_AFL.1
    - Tested as part of the testing for FIA_AFL.1
- Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full)
    - Tested as part of the testing for FAU_STG.1, FAU_STG_EXT.1 and FAU_STG.3
- Ability to configure the cryptographic functionality
    - Tested as part of the testing for FCS_SSHC_EXT.1 and FCS_SSHS_EXT.1
- Ability to set the time which is used for time-stamps
    - Tested as part of the testing for FPT_STM_EXT.1
- Ability to manage the trusted public keys database
    - Tested as part of the testing for FCS_SSHC_EXT.1.9

2.4.6 Restrictions on Security Roles (FMT_SMR.2)

2.4.6.1 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.

Section 6.4.5 of [ST] (“FMT_SMR.2: Restrictions on Security Roles”) states the TOE has three roles: Super User; Admin; and Operator. The Operator role is read-only while the other two roles serve as the Security Administrator for the TOE functions specified in FMT_SMF.1.

2.4.6.2 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.

Section 2.6.2 of [CCECG] (“SSH Configuration”) states the TOE is required to be configured into FIPS mode. [CLI] states that SSH is enabled by default so no additional cryptographic configuration is needed to enable the trusted path beyond enabling FIPS mode. To configure administration authentication methods, section 2.6.3 of [CCECG] (“Supported Authentication Methods”) provides a summary and reference to [CLI] that describes how SSH public key authentication can be used instead of the default username/password. The Super User account is created during initial setup as described in section “Command Line Interface” of [CLI]. Once the TOE has been installed and configured, it will be listening on port 22 on the management port IP address, so no additional client configuration is required.

2.4.6.3 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.

The TOE provides two methods for administering the TOE, both of which have been covered by testing as follows:

- Local administration of TOE components via command line interface

    - covered by testing for FTA_SSL.4 Test 1.
- Remote administration via SSH
    - covered by testing for FTA_SSL.4 Test 2

All remaining testing was performed via one of these interfaces.

2.5 Protection of the TSF (FPT)

2.5.1 Protection of Administrator Passwords (FPT_APW_EXT.1)

2.5.1.1 TSS Activities

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

Section 6.5.1 of [ST] (“FPT_APW_EXT.1: Protection of Administrator Passwords”) states the TOE stores administrative passwords using 256-bit AES and prevents reading of plaintext passwords. It also states the TOE does not offer any functions that will disclose a plaintext password to any users.

2.5.1.2 Guidance Activities

> None defined.

2.5.1.3 Test Activities

> None defined.

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

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 6.2.3 of [ST] (“FCS_CKM.4: Cryptographic Key Destruction”) describes two locations for key storage: plaintext in volatile memory (for which there is no user-facing interface to disclose); and encrypted key storage using 256-bit AES (there is no interface for disclosing this data either, but it is also protected through encryption since the data is persistently stored).

Section 6.5.2 of [ST] (“FPT_SKP_EXT.1: Protection of TSF Data (for Reading of all Pre-shared, Symmetric, and Private Keys”) states the TOE does not offer any functions to disclose stored cryptographic keys.

2.5.2.2 Guidance Activities

> None defined.

2.5.2.3 Test Activities

None defined.

2.5.3 Reliable Time Stamps (FPT_STM_EXT.1)

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 6.5.3 of [ST] ("FPT_STM_EXT.1: Reliable Time Stamps") specifies the TOE's usage of system time to include audit record timestamps, session duration for idle timeout, and for cryptographic operations based on time. Based on a review of the remainder of the TSS, there are no other time functions that need to be considered within the TSF. The reliability of the clock is asserted in section 6.5.3 through a description of the hardware real time clock, which is industry-standard and therefore considered trustworthy for the level of precision required by the TSF.

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

The ST does not select "obtain time from the underlying virtualization system".

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 "Command Line Interface > Root commands > date" of [CLI] instructs the administrator how to use the `date` CLI command to set the date and time. The TOE does not include use of an NTP server in its evaluated configuration.

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 support obtaining time from the underlying virtualization system.

2.5.3.3 Test Activities

The evaluator shall perform the following tests:

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 logged onto the TOE and queried the system time. The evaluator attempted to change the time then queried the time again and verified that the time had changed to the time set by the evaluator.

Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE, and set up a communication path with the NTP server. The evaluator 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's evaluated configuration does not include NTP.

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.

The TOE does not obtain time from the underlying virtualization system.

2.5.4 TSF Testing (FPT_TST_EXT.1)

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 6.5.4 of [ST] ("FPT_TST_EXT.1: TSF Testing") details the software module integrity tests and cryptographic known answer tests the TOE performs at start up. Section 6.5.4 describes how these tests are performed and what they actually do. It provides rationale they are sufficient to demonstrate the TSF is operating correctly as they encompass the TOE's cryptographic functionality and the integrity of the TOE's 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 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 2.6.1 of [CCECG] ("Cryptographic Self-Tests") states the TOE performs a series of self-tests during initial startup. If a self-test fails, the TOE enters an error state where a system recovery prompt is displayed. In this circumstance, the guidance advises the administrator to contact a TippingPoint support representative for assistance. The information in the guidance corresponds to the description provided 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 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 booted the TOE and observed that the self-tests were carried out during initial startup.

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 booted the TOE and observed that the self-tests were carried out during initial startup. As there is no deviation from execution on startup no justification is needed.

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 activity is not applicable.

2.5.5 Trusted Update (FPT_TUD_EXT.1)

2.5.5.1 TSS Activities

The evaluator shall verify that the TSS describe how to query the currently active version.

Section 6.5.5 of [ST] ("FPT_TUD_EXT.1: Trusted Update") states the currently active version can be queried with the `version` CLI command.

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.

The statement of FPT_TUD_EXT.1.1 in Section 5.2.5.5 of [ST] ("Trusted Update (FPT_TUD_EXT.1)") indicates there is no delayed activation capability, so this activity is not applicable.

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 6.5.5 of [ST] states a TOE software update is acquired using the `debug upgrade` CLI command, which takes a download URL as a parameter. The vendor generates a digital signature of the update package by first calculating the SHA-256 hash of the update package, then encrypting the generated hash using its 2048-bit RSA private key. The TOE verifies the digital signature on the update package prior to installing the package. The TOE commences installation only after it has verified the digital signature on the update package and will not install a package that has an invalid signature.

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.

The statement of FPT_TUD_EXT.1.2 in Section 5.2.5.5 of [ST] does not include ‘support automatic checking for updates’ or ‘support automatic updates’. Therefore, this assurance activity is not applicable.

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 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.

The TOE does not use a published hash to verify the integrity of trusted updates, so this activity is not applicable.

2.5.5.2 Guidance Activities

The evaluator shall verify that the guidance documentation describes how to query the currently active version.

Section 2.9 of [CCECG] (“TOE Updates”) states the `show version` CLI command displays the current software 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.

There is no delayed activation function, so this activity is not applicable to the TOE.

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 2.9 of [CCECG] states the TOE updates are initiated manually by the Super User. The integrity of the update is verified prior to installation using a digital signature. TippingPoint Technical Support releases software updates on the Threat Management Center (TMC): https://tmc.tippingpoint.com. The administrator uses the `debug upgrade` CLI command to download a TOE update package directly from a specified URL. The vendor protects package files by first calculating a SHA-256 hash, then signing the hash using a 2048-bit RSA private key. The TOE verifies the digital signature prior to installing the update package. The TOE starts the update process once it verifies the signature. The TOE will not install a package with an invalid signature.

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.

The TOE does not use a published hash to verify the integrity of trusted updates, so this activity is not applicable.

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 activity is not applicable.

If this was information 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 activity is not applicable.

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.

The TOE does not use a certificate-based mechanism for validating the digital signatures of software updates, so this activity is not applicable.

2.5.5.3 Test Activities

The evaluator shall perform the following tests:

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 performed the version verification action and determined the current version of the TOE. The evaluator attempted to install a new version of the TOE software and verified that the TOE successfully installed the TOE. The evaluator queried the current version of the TOE and observed that the TOE reported the current version as the newly installed version.

Test 2 : 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 logged into the TOE and queried the current version. The evaluator then attempted to update the TOE using a modified update, an unsigned update and an update with an invalid signature. The TOE does not support delayed activation of updates. The evaluator confirmed that for each of these attempts the TOE did not accept the update and the TOE's version did not change.

Test 3 : 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.

The TOE does not use a published hash to validate the integrity of trusted updates, so this test is not applicable.

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 TOE does not use a published hash to validate the integrity of trusted updates, so this activity is not applicable.

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).

The evaluator performed Test 1 and Test 2 for manual update, which is the only method supported in the evaluated configuration. The TOE does not use a published hash mechanism, so Test 3 is not applicable and was not performed.

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

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

2.6 TOE Access (FTA)

2.6.1 TSF-initiated Termination (FTA_SSL.3)

2.6.1.1 TSS Activities

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

Section 6.6.1 of [ST] ("FTA_SSL.3: TSF-initiated Termination") states remote administrator sessions by default time out automatically after 15 minutes of inactivity, and an administrator can configure the inactivity timeout to any integer value between 1 and 32,000.

2.6.1.2 Guidance Activities

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

Section "Command Line Interface" of [CLI] states when there has been no CLI activity for 15 minutes, connection to the TOE times out. Note that the TOE does not distinguish between local and remote interactive sessions.

Section "Command Line Interface > Edit running configuration commands > Contexts and related commands > running-aaa Context Commands > ips{running-aaa}login" of [CLI] provides instructions for configuring the inactivity time period for remote administrative session termination. This section also states that the default time period is 15 minutes.

2.6.1.3 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 only method of remote administration of the TOE is remote CLI via SSH.

The evaluator configured the TOE for various inactivity time periods for remote CLI sessions. For each configured value, the evaluator established a remote administrative session with the TOE by logging on to the CLI over SSH, and then ceased activity on the session. The evaluator verified through observation of the interactive session and examination of audit logs the remote session was terminated by the TOE after the configured period of inactivity.

2.6.2 User-initiated Termination (FTA_SSL.4)

2.6.2.1 TSS Activities

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

Section 6.6.2 of [ST] ("FTA_SSL.4: User-initiated Termination") states administrators terminate their own interactive sessions by logging out at the console (local session) and SSH (remote session).

2.6.2.2 Guidance Activities

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

Section "Command Line Interface > Root commands > logout" of [CLI] describes the CLI command used by the administrator to terminate an interactive session (either local or remote).

2.6.2.3 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 evaluator initiated a local interactive session with the TOE via the console port. The evaluator then entered the logout command and confirmed the session was terminated.

> **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 initiated a remote interactive session with the TOE via the network management port using SSH. The evaluator then entered the logout command and confirmed the session was terminated.

2.6.3 TSF-initiated Session Locking (FTA_SSL_EXT.1)

2.6.3.1 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.

Section 6.6.3 of [ST] ("FTA_SSL_EXT.1: TSF-initiated Session Locking") states local administrator sessions can be configured to time out after a period of inactivity. The inactivity timeout period is specified in minutes and can be set to any integer value between 1 and 32,000.

2.6.3.2 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 “Command Line Interface” of [CLI] states when there has been no CLI activity for 15 minutes, connection to the TOE times out. Note that the TOE does not distinguish between local and remote interactive sessions.

Section “Command Line Interface > Edit running configuration commands > Contexts and related commands > running-aaa Context Commands > ips{running-aaa}login” of [CLI] provides instructions for configuring the inactivity time period for remote administrative session termination. This section also states that the default time period is 15 minutes.

2.6.3.3 Test Activities

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 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 evaluator configured the TOE for various inactivity time periods for local CLI sessions. For each configured value, the evaluator established a local administrative session with the TOE by logging on to the CLI, and then ceased activity on the session. The evaluator verified through observation of the interactive session that the local session was terminated by the TOE after the configured period of inactivity.

2.6.4 Default TOE Access Banners (FTA_TAB.1)

2.6.4.1 TSS Evaluation Activity

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 6.3.4 of [ST] (“FIA_UIA_EXT.1: User Identification and Authentication, FIA_UAU_EXT.2: Password-based Authentication Mechanism”) describes the methods of access available to an administrator. The TOE provides local access to its CLI via its console interface or direct connection to the Ethernet Management port, and remote access to the CLI via the Ethernet Management port using SSH. Section 6.6.4 of [ST] (“FTA_TAB.1: Default TOE Access Banners”) states that a configurable access banner is displayed at both the local and remote instances of the CLI prior to administrator authentication being completed. The same banner is displayed for both interfaces.

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 "Command Line Interface > Edit running configuration commands > Contexts and related commands > running-aaa Context Commands > ips{running-aaa}login-banner" of [CLI] describes the CLI command used by the administrator to configure the banner message.

2.6.4.3 Test Activities

The evaluator shall also perform the following test:

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 a notice and consent warning message. The evaluator verified the configured message was displayed when the evaluator connected to the TOE via the console interface (local access). In this case, the TOE displays the banner message before displaying the login prompt. The evaluator verified the configured message was displayed when the evaluator connected to the TOE via the Ethernet Management port using SSH. In this case, the TOE displays the banner message prior to prompting for the user password (when password-based authentication is used) or prior to displaying the CLI command prompt (when public key-based authentication is used).

2.7 Trusted Path/Channels (FTP)

2.7.1 Inter-TSF Trusted Channel (FTP_ITC.1)

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.

The requirement identifies audit server as the only authorized IT entity with which the TOE communicates over a trusted channel. Section 6.7.1 of [ST] ("FTP_ITC.1: Inter-TSF Trusted Channel") states the TOE uses SSH to protect communications between itself and the audit server and that the TOE initiates communication with the audit server. Section 6.2.9 of [ST] ("FCS_SSHC_EXT.1 – SSH Client Protocol / FCS_SSHS_EXT.1 – SSH Server Protocol") states the TOE acts as an SSH client for secure communication with an external audit server. Section 6.7.1 of [ST] states SSH provides assured identification of the non-TSF endpoint via association of the host name with its public key. The TSS description of the protected communication between the TOE and the external audit server is sufficiently detailed to be able to match it with FCS_SSHC_EXT.1 specified in section 5.2.2.9 of [ST] ("SSH Client Protocol (FCS_SSHC_EXT.1)").

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 “SSH configuration” of [CLI] states the TOE can be configured to send syslog messages over SSH using the “Remote System Log” contact.

Section “Edit running configuration commands > Contexts and related commands > running-gen Context Commands > ips{running-gen}ssh” of [CLI] states when an SSH connection to a remote syslog breaks, the device automatically attempts to reconnect three times over the course of a minute (once every 20 seconds for one minute). Each failed attempt is logged locally, and if the connection is still broken after one minute, the device stops attempting to reconnect. If the connection is broken and the automatic attempt to reconnect fails, the administrator must disable the connection and then re-enable the "Remote System Log" configuration. Any data that was queued before the connection was lost gets sent after the connection is re-established. All data is sent in real time.

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.

The evaluator shall perform the following tests:

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

The evaluator tested the TOE’s ability to communicate with an external audit server using SSH while testing FAU_STG_EXT.1.

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.

While testing the TOE’s ability to communicate with an external audit server using SSH, the evaluator confirmed the communication channel is initiated by the TOE.

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

While testing the TOE’s ability to communicate with an external audit server, the evaluator confirmed all such communication occurs over SSH and that no channel data is sent in plaintext.

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

The evaluator shall, for each instance where the TOE acts as a client utilizing a secure communication mechanism with a distinct IT entity, 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 two separate SSH connections from the TOE to the audit server by physically unplugging the network cable, then plugging the cable back in at intervals greater and less than the application layer timeout setting. The physical interruption was performed at a core level switch which facilitates the connection of the TOE to the remote device. The evaluation team confirmed through viewing of log messages and packet captures that each connection was re-established and that no data was communicated unprotected.

Further assurance activities are associated with the specific protocols.

Refer to the testing for FCS_SSHC_EXT.1 in section 2.2.9.3 above as the TOE utilizes SSH as a client for this purpose.

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 activity is not applicable.

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

The developer provided the application layer configuration settings for the TOE's SSH client implementation, which enabled the evaluation team to complete the test activities specified for FTP_ITC.1.

2.7.2 Trusted Path (FTP_TRP.1/Admin)

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 6.7.2 of [ST] ("FTP_TRP.1/Admin: Trusted Path") states the TOE protects communications with remote administrators accessing the CLI using SSH. Remote administrators initiate communication via the trusted path by using an SSH client to login. To successfully establish an interactive administrative session, the administrator must be able to provide acceptable user credentials, after which they will be able to access the CLI features. This is consistent with the protocols specified in the requirement. Furthermore, the ST includes FCS_SSHS_EXT.1 to specify the TOE's SSH server functionality.

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 "Command Line Interface" of [CLI] contains instructions for establishing a remote administrative session with the CLI using SSH.

2.7.2.3 Test Activities

The evaluated shall perform the following tests.

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.

The evaluator set up the SSH trusted path connection as specified in [CLI].

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

The evaluator confirmed the SSH connection was established successfully with the TOE and that all communicated data was protected.

3 Security Assurance Requirements

3.1 Class ASE: Security Targeted 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

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

Note that additional Evaluation Activities for the TSS in the case of a distributed TOE are defined in section A.9.1.1 in [CPP_ND_V2.2-SD].

The TOE is not a distributed TOE. Therefore, this activity is not applicable.

3.2 Class ADV: Development

3.2.1 ADV_FSP.1 Basic Functional Specification

The EAs for this assurance component focus on understanding the interfaces (e.g., application programming interfaces, command line interfaces, graphical user interfaces, network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in Section 2, and in EAs for AGD, ATE and AVA SARs in other parts of Section 3.

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.

Through review of [CCECG] and [CLI], the evaluation team identified that the following external interfaces are security relevant:

- SSH logical interface
- Command-line interface
- Syslog interface.

The evaluation team determined the interface documentation described the purpose and method of use for each TSFI identified as being security relevant, sufficient to enable each of the guidance assurance activities to be completed satisfactorily. The evaluation team's results from performing the guidance assurance activities are documented in Sections 2 and 3 of this AAR.

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 evaluation team determined the interface documentation identified and described the parameters for each TSFI identified as being security relevant, sufficient to enable each of the guidance assurance activities to be completed satisfactorily. The evaluation team's results from performing the guidance assurance activities are documented in Sections 2 and 3 of this AAR.

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'.

In performing the guidance activities specified for each of the SFRs claimed in [ST], the evaluation team examined the interface documentation presented in [CLI] and [CCECG]. The evaluation team was able to perform all the guidance assurance activities, identifying the interfaces relevant to each SFR in the process. The evaluation team's results from performing the guidance assurance activities are documented in Sections 2 and 3 of this AAR.

3.3 Class AGD: Guidance Documents

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

Note that additional Evaluation Activities for the guidance documentation in the case of a distributed TOE as defined in section A.9.1.1 in [CPP_ND_V2.2-SD].

The TOE is not a distributed TOE. Therefore, this activity is not applicable.

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.

In addition, the evaluator performs the EAs specified below.

3.3.1.1 AGD_OPE.1 Evaluation Activity

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

All operational guidance documentation for the TOE is available from the Online Help Center at https://docs.trendmicro.com/en-us/documentation/threat-protection-system/. References to the Online Help Center appear in customer emails, product announcements, Release Notes, etc.

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.

[ST] does not describe multiple Operational Environments for the TOE so this part of the evaluation activity is not applicable. Section 1.1 of [ST] ("Security Target, TOE and CC Identification") identifies the following platforms for the TOE:

- Hardware appliances:
  - TPS 1100TX
  - TPS 5500TX
  - TPS 8200TX
  - TPS 8400TX
  - TPS 8600TXE
  - TPS 9200TXE
- Virtual appliance
  - vTPS.

The TOE hardware appliances are clearly identified in [HSIG]. As the TOE functional behavior is identical across hardware models, [CLI] does not draw a distinction between individual hardware appliances, since there are no security-relevant differences between them.

The TOE virtual appliance (vTPS) is clearly identified in [vTPSUG]. Section "vTPS Functionality > Unsupported features" identifies features supported in the hardware appliances that are not supported by vTPS, while section "vTPS Functionality > Commands" lists CLI commands supported by the hardware appliances that are not supported by vTPS. Otherwise, [CLI] is equally applicable to vTPS as it is to the hardware appliances.

Section 2 of [CCECG] ("Configuration for Common Criteria") identifies the Trend Micro TippingPoint Threat Protection System hardware appliances and I/O modules included in the TOE and describes the evaluated operational environment for the vTPS.

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 2.4 of [CCECG] ("Configuring the TPS for Common Criteria Compliance") states the TOE must be configured to support the Federal Information Processing Standards 140-2 (FIPS 140-2) cryptographic requirements. The FIPS-CC Mode restricts the cryptographic mechanisms to FIPS-approved algorithms.

Section 2.4 of [CCECG] also warns the administrator the cryptographic engine used when the TPS has been placed into FIPS mode is used for all SSH and other cryptographic functionality within the scope of the evaluated configuration of the product and no other cryptographic engine or configuration was evaluated or tested during the Common Criteria evaluation of TPS.

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 2.2 of [CCECG] ("Scope of Evaluation") states the evaluated functionality is scoped exclusively to the security functional requirements specified in [ST]. In particular, the SSH protocol implemented by the

Trend Micro TippingPoint devices has been tested, and only to the extent specified by the security functional requirements.

3.3.1.5 AGD_OPE.1 Evaluation Activity

Modified in accordance with 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 2.4 of [CCECG] ("Configuring the TPS for Common Criteria Compliance") states the TOE must be configured to support the Federal Information Processing Standards 140-2 (FIPS 140-2) cryptographic requirements. The FIPS-CC Mode restricts the cryptographic mechanisms to FIPS-approved algorithms.

Section 2.9 of [CCECG] ("TOE Updates") states TOE updates are verified by a digital signature. Instructions are provided on how to download a TOE update package directly from a specified URL.

Section 2.2 of [CCECG] ("Scope of Evaluation") states the evaluated functionality is scoped exclusively to the security functional requirements specified in [ST]. In particular, the SSH protocol implemented by the Trend Micro TippingPoint devices has been tested, and only to the extent specified by the security functional requirements.

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).

Section 2.3 of [CCECG] ("Operating Environment Assumptions") includes a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target).

The following operating environment requirements must be met for Common Criteria operation:

- The TOE is to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains.
- The TOE provides networking functionality as its core function and does not provide functionality/services that could be considered general-purpose computing.
- The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
- The TOE administrators are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device.
- The TOE's firmware and software is to be updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
- TOE administrators must ensure there is no unauthorized access possible to sensitive residual information (e.g., cryptographic keys, keying material, PINs, passwords) on networking equipment when the equipment is discarded or removed from its operational environment.
- For virtualized deployments, administrators of the virtualization system are assumed to be trusted and to act in the best interest of security for the organization. This includes not interfering with the correct operation of the device.
- Virtualization system software is assumed to be updated by the administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
- For virtualized deployments, it is assumed the virtualization system provides, and is configured to provide, sufficient isolation between software running in virtual machines on the same physical platform. Furthermore, it is assumed that the virtualization system adequately protects itself from software running inside virtual machines on the same physical platform.

- For virtualized deployments, it is assumed that the virtualization system and virtual machines are correctly configured to support TOE functionality implemented in virtual machines.

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.

[ST] does not describe multiple Operational Environments for the TOE so this part of the evaluation activity is not applicable. Section 1.1 of [ST] ("Security Target, TOE and CC Identification") identifies the following platforms for the TOE:

- Hardware appliances:
  - TPS 1100TX
  - TPS 5500TX
  - TPS 8200TX
  - TPS 8400TX
  - TPS 8600TXE
  - TPS 9200TXE
- Virtual appliance:
  - vTPS.

The TOE hardware appliances are clearly identified in [HSIG]. As the TOE functional behavior is identical across hardware models, [CLI] does not draw a distinction between individual hardware appliances, since there are no security-relevant differences between them.

The TOE virtual appliance (vTPS) is clearly identified in [vTPSUG]. Section "vTPS Functionality > Unsupported features" identifies features supported in the hardware appliances that are not supported by vTPS, while section "vTPS Functionality > Commands" lists CLI commands supported by the hardware appliances that are not supported by vTPS. Otherwise, [CLI] is equally applicable to vTPS as it is to the hardware appliances.

Section 2 of [CCECG] ("Configuration for Common Criteria") identifies the Trend Micro TippingPoint Threat Protection System hardware appliances and I/O modules included in the TOE and describes the evaluated operational environment for the vTPS.

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.

[ST] does not describe multiple Operational Environments for the TOE. However, the TOE comprises both hardware and virtual appliances. [vTPSUG] provides instructions to install the vTPS virtual appliance, while [HSIG] provides instructions for installing the TOE hardware appliances.

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.

Section 2.4 of [CCECG] (“Configuring the TPS for Common Criteria Compliance”) provides instructions for managing the security of the TOE both as a product and as a component of the larger operational environment.

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.

Section 2.4 of [CCECG] (“Configuring the TPS for Common Criteria Compliance”) states the password of the initial Super User account must be set on first use. This section also describes how SSH is enabled by default to provide cryptographically protected remote administration. Section 2.8 of [CCECG] (“Security Management”) defines the different administrator roles and the privileges available to them, which indicates to readers how they can grant administrative access to the TOE on a least-privilege basis.

3.4 Class ALC: Life-Cycle Support

3.4.1 ALC_CMC.1 Labelling of the TOE

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

3.4.2 ALC_CMS.1 TOE CM Coverage

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

3.5 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.

The evaluator should consult Appendix A [in [CPP_ND_V2.2-SD]] when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

Testing of the TOE was performed at the Leidos Accredited Testing and Evaluation Lab located in Columbia, Maryland from December 2023 to August 2024. Some additional testing of SSH was performed in December 2024.

The evaluation team established a test configuration comprising:

- 1100TX

- vtpsESXi on VMware ESXi 7.0
- vtpsKVM on RHEL 8.9

The test configuration included the following devices in the operational environment of the TOE:

- Ubuntu Linux host to facilitate SSH testing
- Ubuntu Linux host for hosting of Apache server to provide update images
- Kali Linux host facilitating test scripts
- 2x VMware ESXi host to host Virtual Machines used in testing.

Each relevant Test Activity identified in [CPP_ND_V2.2-SD] was given its own test case in the Test Report. Each test case consists of the test steps specified in the Supporting Document, along with the actual test steps performed by the evaluators and any corresponding evidence. Additional information on the set up of the test environment and use of the additional test equipment can also be found in each test case.

Note that additional Evaluation Activities relating to evaluator testing in the case of a distributed TOE are defined in section A.9.3.1 in [CPP_ND_V2.2-SD].

The TOE is not a distributed TOE, so these additional activities are not applicable.

3.6 Class AVA: Vulnerability Assessment

3.6.1 AVA_VAN.1 Vulnerability Survey

While vulnerability analysis is inherently a subjective activity, a minimum level of analysis can be defined and some measure of objectivity and repeatability (or at least comparability) can be imposed on the vulnerability analysis process. In order to achieve such objectivity and repeatability it is important that the evaluator 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 in [CPP_ND_V2.2-SD], while an “outline” of the assurance activity is provided below.

3.6.1.1 AVA_VAN.1 Evaluation Activity (Documentation)

Modified in accordance with 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[1] 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) such as a web server and 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 hypotheses during their analysis.

The vendor provided information on the hardware and software components of the TOE. Specifically, [ST] identifies the model names of TOE, the OS version used (Linux-5.4.58-yocto-standard), the processors used, and the cryptographic library used (OpenSSL 3.0.9). The vendor separately provided proprietary material on specific third-party libraries used by the TOE, including version information. The hardware appliances included in the TOE and their specific processors are as follows:

- TPS 1100TX—Intel Pentium D-1517 (Broadwell microarchitecture)
- TPS 5500TX—Intel Pentium D-1559 (Broadwell microarchitecture)
- TPS 8200TX—Intel Xeon E5-2648L v3 (Haswell-EP microarchitecture)
- TPS 8400TX—Intel Xeon E5-2648L v3 (Haswell-EP microarchitecture)
- TPS 8600TXE—Intel Xeon Gold 5318N (Ice Lake microarchitecture)
- TPS 9200TXE—Intel Xeon Gold 5318N (Ice Lake microarchitecture).

This information was used as inputs to the vulnerability analysis, in addition to variations on the TOE's name (e.g., “TippingPoint”, “TPS”, and “Threat Protection System”).

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 [CPP_ND_V2.2E, 3.4]

b) a mapping of the auditable events recorded by each distributed TOE component as in [CPP_ND_V2.2E, 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.

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

[1] In this sub-section the term “components” refers to parts that make up the TOE. It is therefore distinguished from the term “distributed TOE components”, which refers to the parts of a TOE that are present in one physical part of a distributed TOE. Each distributed TOE component will therefore generally include a number of the hardware and software components that are referred to in this sub-section: for example, each distributed TOE component will generally include hardware components such as processors and software components such as an operating system and libraries.

The evaluation team performed a search of the following public vulnerability databases:

- National Vulnerability Database (https://nvd.nist.gov/)
- US-CERT Vulnerability Notes Database (https://www.kb.cert.org/vuls/)
- Tipping Point Zero Day Initiative (https://www.zerodayinitiative.com/advisories/published/).

Searches were performed several times, most recently on 4 December 2024, using search terms that referenced the TOE itself, the processors that the physical TOE models use, the OS kernel version, the cryptographic library, and the list of additional third-party software components provided by the vendor.

No vulnerabilities were identified for the TOE.

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