Assurance Activity Report for
Trellix Intrusion Prevention System Sensor and Manager Appliances
version 11.1

Trellix Intrusion Prevention System Sensor and Manager Appliances version 11.1 Security Target

collaborative Protection Profile for Network Devices
Version 2.2e,
PP-Module for Intrusion Protection Systems (IPS), Version 1.0

AAR Version 1.4, May 14 2024

Evaluated by:

2400 Research Blvd, Suite 395
Rockville, MD 20850

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:
Trellix

The Author of the Security Target:
Acumen Security, LLC

The TOE Evaluation was Sponsored by:
Trellix

Evaluation Personnel:
Yogesh Pawar
Pratheek Menon
Sagar Pujari
George Kumi

Acumen Security LLC

Common Criteria Version
Common Criteria Version 3.1 Revision 5

Common Evaluation Methodology Version
CEM Version 3.1 Revision 5

Revision History

VERSION	DATE	CHANGES
1.0	16/01/2024	Initial Release
1.1	02/04/2024	Updated to address lead review comments
1.2	15/04/2024	Minor Updated based on internal reviews
1.3	08/05/2024	Updated as per ECR comments
1.4	14/05/2024	Updated as per 2nd round of ECR comments

Contents

1 TOE Overview

The TOE is comprised of the Trellix Intrusion Prevention System (IPS) software running on one Trellix Intrusion Prevention System Manager Appliance and one or more Trellix Intrusion Prevention System Sensor (Sensor).

The Trellix Intrusion Prevention System (IPS) Sensor performs stateful inspection on a per-packet basis to discover and prevent intrusions, misuse, denial of service (DoS) attacks, and distributed denial of service (DDoS) attacks. Trellix Intrusion Prevention System (IPS)is available in multiple Sensor appliances providing different bandwidth and deployment strategies.

Trellix IPS Manager (IPS Manager) is used to manage, push configuration data and policies to the Sensors. Communication between Manager and Sensors uses secure channels that protect the traffic from disclosure and modification. Authorized administrators may access the Manager via a GUI (over HTTPS) or a CLI (via SSH or a local connection). Sensors may be accessed via CLI (via SSH or a local connection) for initial setup. Once initial setup is complete, all management occurs via the Manager.

The Sensor's presence on the network is transparent. The Sensor is protected from the monitored networks as the system is configured to not accept any management requests or input from the monitored networks.

1.1.1 IPS Manager Architecture

The Manager Appliance is management console of the Trellix Intrusion Prevention System (IPS). The Manager Appliance is a 1-U rack dense chassis with multi-core Intel XEON Series Processor. The Manager Appliance runs on a pre-installed, hardened MLOS operating system and comes pre-loaded with the IPS Manager software. Manager is used, to manage, push configuration data and policies to the Sensors.

1.1.2 Sensor Architecture

The primary function of the Sensor (also referred to as the Collector Component) is to analyze traffic on selected network segments and respond when an attack is detected. The Sensor examines the header and data portion of every network packet; scanning for patterns and behavior in the network traffic that indicates malicious activity.

The Sensor can operate in three modes:

1) **Inline**: The product is installed as an appliance within the network that applicable traffic must flow through.

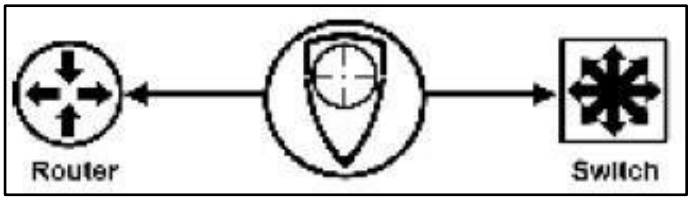


Figure 1: Sensor in 'Inline mode'

2) **Tap:** The network traffic flows between the clients and servers, and the data is copied by the tap to the Sensor, which is essentially invisible to the other network entities. Note that the TOE cannot inject response packets back through an external tap, so Sensors offer response ports through which a response packet (such as a TCP reset) can be injected to close a malicious connection.

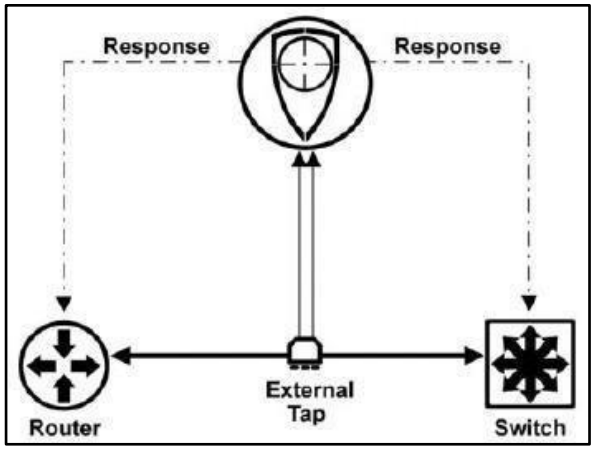


Figure 2: Sensor in 'Tap' mode

3) **Span**: The traffic is spanned off either the server side or the client side of a router or switch, copying both the incoming and outgoing traffic from any one of the ports. This requires a special network device that has a span port capability. Note that SPAN mode is also a "sniffing" mode, which—unlike inline mode—does not enable the TOE to prevent attacks from reaching their targets. However, while the TOE can issue response packets via the Sensor's response ports, some switches allow response packets to be injected by an IPS back through the SPAN port.

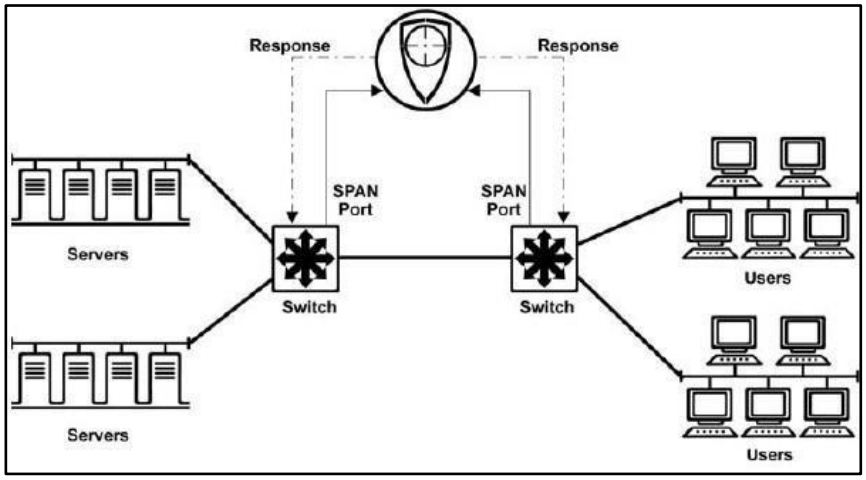


Figure 3: Sensor in 'Span' mode

A single multi-port Sensor can monitor many network segments in any combination of operating modes: monitoring or deployment mode for the Sensor; SPAN mode, TAP mode, or INLINE mode.

The IPS's Virtual IDS (VIDS) feature enables users to further segment a port on a Sensor into many "Virtual Sensors". A VIDS can be dedicated to a specific network port with monitoring rules appropriate for that segment. These rules may be different than the rules used to monitor other segments.

Alternately, if a monitored network segment includes the use of Virtual LANs (VLANs) or Classless Inter- Domain Routing (CIDR), one or more VIDS can be directed at monitoring them, with VIDS each configured with distinct monitoring rules. Note that VIDS are not particularly security relevant in and of themselves, but rather serve to organize and distinguish monitoring rules.

2 Assurance Activities Identification

The Assurance Activities contained within this document include all those defined within the NDcPP_v2.2e based upon the core SFRs and those implemented based on selections within the PP.

The Assurance Activities contained within this document include all those defined within the NDcPP_v2.2e and MOD_IPS_V1.0 based upon the core SFRs and those implemented based on selections within the PP.

3 Test Equivalency Justification

3.1 Processors:

There are numerous form factors for the IPS Sensors that vary in connections, storage, and memory. They can be grouped into two categories of CPUs: Intel XEON GOLD, and Intel ATOM. Intel XEON and Intel ATOM processors are inherently not equivalent processors; therefore, processor equivalency considers XEON and ATOM processors separately. The following tables summarize the security-relevant characteristics of the processors used in the IPS Sensors.

Table 1: TOE platforms

TOE Model	Description	Analysis
Operating System – This is the OS that runs on the platform		
IPS-NS9500	Sensor OS 11	The TOE is a purpose-built security appliance. The TOE does not run a general-purpose operating system. This OS does not give users access to underlying general-purpose functionality. In total, there are 7 different series with multiple models in each series. Out of which 5 models are Sensors and 2 Models are Managers
IPS-NS7600	Sensor OS 11	
IPS-NS7500	Sensor OS 11	
IPS-NS3600	Sensor OS 11	
IPS-NS3200	Sensor OS 11	
NSM-MAPL-NG	MLOS 3	
NSM-MAPL-NG	MLOS 3	
Base Processor/Route Processor		
IPS-NS9500	2 x XEON GOLD 6230	All Manager models are running on Intel XEON processors. Across all Sensors models, there are 2 different processors, Intel XEON and Intel ATOM.
IPS-NS7600	1 x XEON SILVER 4416+	
IPS-NS7500	1 x XEON GOLD 5218N	
IPS-NS3600	1 x XEON D-1734NT	
IPS-NS3200	1 x ATOM C2538	
NSM-MAPL-NG	1 x XEON SILVER 4210	
NSM-MAPL-NG	1 x XEON SILVER 4114	

Table 2: Intel XEON Processor Security-Relevant Characteristics

Processor Item	GOLD 6230	SILVER 4416+	GOLD 5218N	D-1734NT
Micro-architecture	Scalable (Cascade Lake)	Scalable (Sapphire Rapids)	Scalable (Cascade Lake)	Scalable (Ice Lake)
Instruction Set Extensions (ISEs)	SSE4.2, AVX, AVX2, AVX-512	AMX, SSE4.2, AVX, AVX2, AVX-512	SSE4.2, AVX, AVX2, AVX-512	AVX-512
AES New Instructions	Yes	Yes	Yes	Yes
Secure Key	No	No	No	No
OS Guard	No	Yes	No	No
Trusted Execution Technology	Yes	Yes	Yes	Yes
Execute Disable Bit	Yes	Yes	Yes	Yes

Two of the Intel XEON processors are on the same micro-architecture (Cascade Lake) and are hence considered equivalent. The others have different microarchitectures, and a deeper inspection of the security-relevant processor characteristics shows that the differences are:

- Differing ISEs, with the aggregate set consisting of AMX, AVX, AVX2, AVX-512, and SSE4.2.
- Differing support for OS Guard.

Based on this analysis, all the XEON processors cannot be considered equivalent, Hence, three models will be tested.

Table 3: Intel ATOM Processor Security-Relevant Characteristics

Processor Item	ATOM C2538
Micro-architecture	Rangeley
Instruction Set Extensions (ISEs)	None
AES New Instructions	Yes
Secure Key	No
OS Guard	No
Trusted Execution Technology	No
Execute Disable Bit	Yes

Here one model from ATOM CPU will be tested.

Result: One model from each different microarchitecture will be required to be tested. Three models from Intel XEON Processors, One Model from Intel ATOM Processors and two Manager devices with Cascade Lake and Skylake Processors will be tested.

3.2 Software/OS Dependencies:

The source code for the TSF is largely identical across all sensor models. Differences are for scalability issues such as the number of monitoring interfaces supported and the size of internal tables.

3.3 Differences in Libraries Used to Provide TOE Functionality:

There is no difference in the TOE libraries that provide TOE functionality between IPS Sensors. The TOE leverages third party software which is the same for all models.

3.4 TOE Management Interface Differences:

TOE management interface on the IPS Sensors is via remote or local access to the console port. Remote access is secured via SSHv2. The management interfaces are the same across all IPS Sensors.

3.5 TOE Functional Differences:

There are no functional differences between the IPS Sensors. The description below details the TOE's security functionality and which component performs this functionality.

- **Security Audit**

The TOE generates audit records related to TOE operation and administration. These audit records are stored on the Manager platform (and stored in a local database) and are also forwarded to an external audit server. The database stores 50,000 audit records. When the database reaches capacity, the oldest audit records are overwritten. The IPS Sensor generates audit records and forwards the audit records to the Manager platform. If the Manager platform is not available, The IPS Sensor caches audit records in a local file. When connectivity with Manager is restored, the file is uploaded and then deleted. If the file reaches capacity, new events are dropped. Only authenticated users can view audit records.

- **Cryptographic Support**

The TOE provides cryptography in support of other TOE security functionality. All the algorithms claimed have three CAVP certificates (Operational Environments: Intel XEON GOLD, Intel Xeon Silver, and Intel ATOM). The TOE uses the Trellix Intrusion Prevention System cryptographic module.

The TOE provides cryptography in support of secure communications between

  - Sensor and the Manager using TLS.
  - Sensor and the SCP Server (for firmware update) are secured using SSH.
  - Manager and the Audit Server (for audit record upload) are secured using TLS.
  - Management workstation and TOE are secured using SSH or HTTPS.

- **Identification and Authentication**

Administrators connecting to the TOE are required to enter an administrator username and password to authenticate the administrative connection prior to access being granted.

The Manager and IPS Sensors authenticate to one another through a shared secret that is configured during the initial installation and setup process of the TOE. Individual Sensors may use self-signed certificates or CA-signed certificates. The Manager supports both self-signed and CA-signed certificates simultaneously for communication with multiple Sensors.

- **Security Management**

An administrative CLI can be accessed via the Console port or SSH connection, and an administrative GUI on the Manager may be accessed via HTTPS. These interfaces are used for administration of the TOE, including audit log configuration, upgrade of firmware and signatures, administration of users, configuration of SSH and TLS connections.
Only administrators authenticated to the “Admin” role are considered to be authorized administrators.

- **Protection of the TSF**

The IPS Sensors components presence on the network is transparent (other than network packets sent as reactions to configure IPS conditions). The IPS Sensors are protected from the monitored networks as the system is configured to not accept any management requests or input via the monitored interfaces.
The TOE users must authenticate to the TOE before any administrative operations can be performed on the system.
The TOE ensures consistent timestamps are used by synchronizing time information on the IPS
Sensors with the Manager, so that all parts of the IPS system share the same relative time information.
Synchronization occurs over a secure communications channel. Time on the Manager may be configured by an administrator.
The administrator can query the currently installed versions of software on the TOE components using the “show” command, which returns details of the software and hardware version. Trusted update of the TOE software can be performed from the Manager UI, which is then pushed out to the Sensors.
A suite of self-tests is performed by the TOE at power on and conditional self-tests are performed continuously.

- **TOE Access**

The TOE monitors local and remote administrative sessions for inactivity and terminates the session when a threshold time is reached. An advisory notice is displayed at the start of each session.

- **Trusted Path/Channels**

The TSF provides the following trusted communication channels:

  - TLS for an audit server
  - TLS for communication between Manager and Sensors
  - SSH for communication with an SCP Server at Trellix for updates

The TOE implements TLS/HTTPS and SSH for protection of communications between itself and the administrators.

- **Intrusion Prevention**

The TOE performs analysis of IP-based network traffic and detects violations of administratively defined IPS policies. The TOE inspects each packet header and payload for anomalies and known signature-based attacks and performs configured actions for policy violations.
IPS functionality is common across all IPS Sensor appliances. No cryptographic operations are used to perform IPS processing.

3.6 Equivalency Conclusions:

Based on the equivalency rationale above, multiple sensors must be tested because they are considered non-equivalent for cryptographic operations. In addition, two Manager platforms must be tested because they are non-equivalent for cryptographic operations. Therefore, base testing for NDcPP will be performed on the following six platforms:

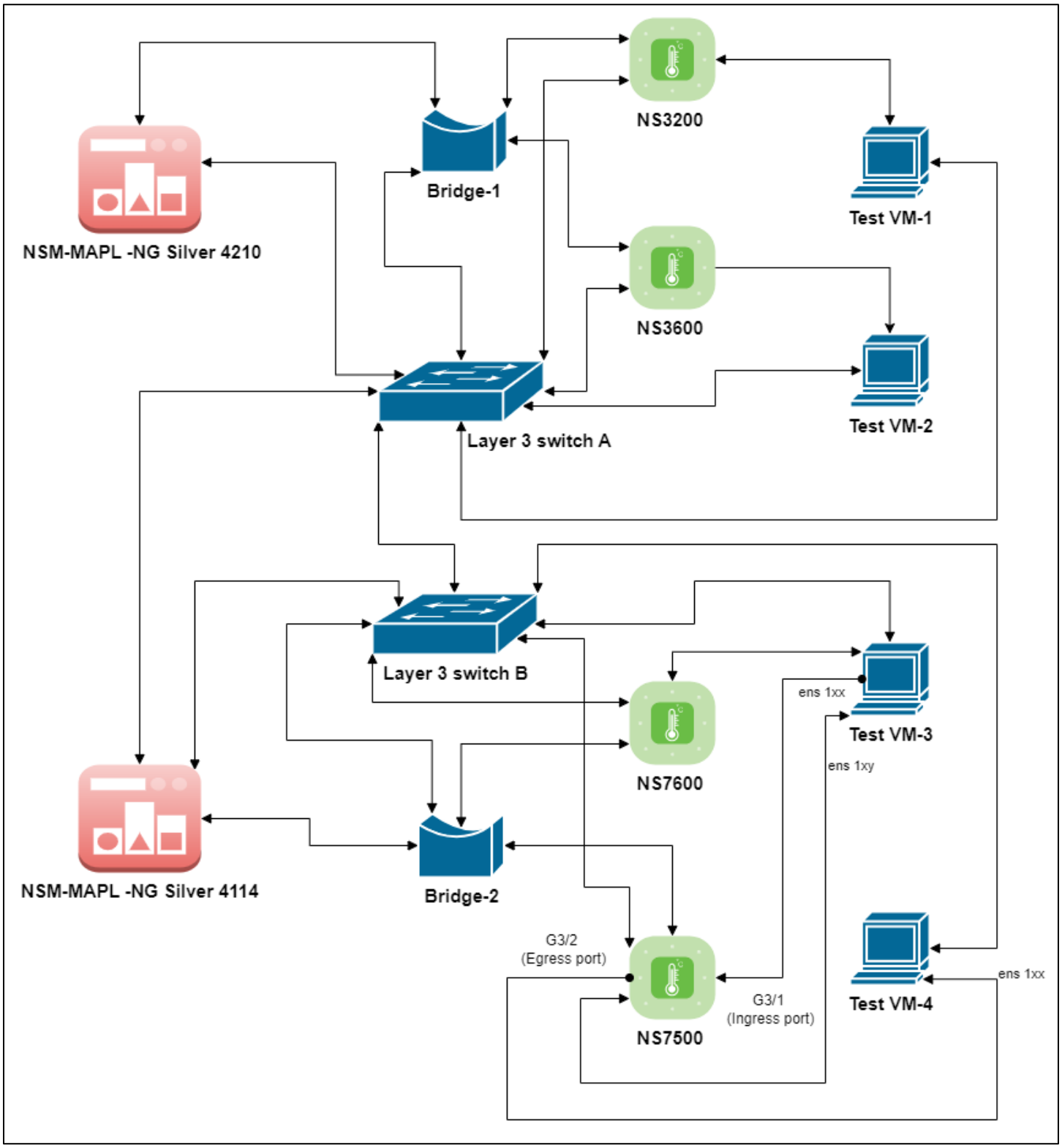
- IPS Sensors:
  - IPS-NS7600 (Sensor with Sapphire Rapids processor)
  - IPS-NS7500 (representative of Intel XEON-based platforms with Cascade Lake micro-architecture)
  - IPS-NS3600 (Sensor with Ice Lake processor)
  - IPS-NS3200 (representative of Intel ATOM C2538-based platforms)
- Manager:
  - NSM-MAPL-NG (Manager with Cascade Lake Processor)
  - NSM-MAPL-NG (Manager with Skylake Processor)

This list of platforms includes both components of the TOE (Manager and IPS Sensor). It also includes one IPS Sensor from each CPU grouping described above.
IPS testing is differentiated from base NDcPP testing for the following reasons:

- IPS processing is only performed on the IPS Sensors
- All IPS code is written in processor-agnostic form for which no processor specific optimization is used (e.g, if recompiled as a separate binary all IPS code would run identically on all of the Sensors with no recompilation needed)
- As stated previously, performance-related processor extensions have no impact on IPS other than faster execution, which is not security-relevant
- The most significant processor difference outside of performance extensions relates to cryptographic processing. IPS functionality does not rely on any cryptographic functionality so those differences are not relevant.

Therefore, all sensor platforms are considered equivalent for IPS functional testing, and this testing can be performed on a single representative platform. This testing will be performed on the IPS-NS7500.

4 Test Bed Descriptions



4.1 Test Time and Location

All testing was carried out at the Acumen Security offices located at 2400 Research Blvd Suite #395, Rockville, MD 20850. Testing occurred from 27th May 2023 to 20th March 2024.
The TOE was in a physically protected, access controlled, designated test lab with no unattended entry/exit ways. At the start of each day, the test bed was verified to ensure that it was not compromised. All evaluation documentation was always kept in a secure repository.

4.2 Configuration Information

Name	OS	Version	Function	Protocols	Time	Tools (version)
NSM-MAPL -NG Silver 4210	MLOS 3.0	11.1.19.3	TOE	TLS/SSH	Manually Set and Verified	N/A
NSM-MAPL -NG Silver 4114	MLOS 3.0	11.1.19.3	TOE	TLS/SSH	Manually Set and Verified	N/A
NS3200	MLOS 3.0	11.1.17.2	TOE	TLS/SSH	Manually Set and Verified	N/A
NS3600	MLOS 3.0	11.1.17.2	TOE	TLS/SSH	Manually Set and Verified	N/A
NS7500	MLOS 3.0	11.1.17.2	TOE	TLS/SSH	Manually Set and Verified	N/A
NS7600	MLOS 3.0	11.1.17.2	TOE	TLS/SSH	Manually Set and Verified	N/A
Test VM-1	Ubuntu	20.04.6	Test Workstation	TLS /SSH	Manually Set and Verified	OpenSSL (1.1.1f) OpenSSH(8.2p1) tcpdump (4.9.3) hexedit (1.4.2) acumen-tlss tool acumen-tlsc tool acumen-sshs tool acumen-sshc tool
Test VM-2	Ubuntu	20.04.6	Test Workstation	TLS /SSH	Manually Set and Verified	OpenSSL (1.1.1f) OpenSSH(8.2p1) tcpdump (4.9.3) hexedit (1.4.2) acumen-tlss tool acumen-tlsc tool acumen-sshs tool

Name	OS	Version	Function	Protocols	Time	Tools (version)
						acumen-sshc tool
Test VM-3	Ubuntu	20.04.6	Test Workstation	TLS /SSH	Manually Set and Verified	OpenSSL (1.1.1f) OpenSSH(8.2p1) tcpdump (4.9.3) hexedit (1.4.2) Scapy (2.4.4) Hydra (9.0) Nmap (7.80) acumen-tlss tool acumen-tlsc tool acumen-sshs tool acumen-sshc tool
Test VM-4	Ubuntu	20.04.6	Test Workstation	TLS /SSH	Manually Set and Verified	OpenSSL (1.1.1f) OpenSSH(8.2p1) tcpdump (4.9.3) hexedit (1.4.2) Scapy (2.4.4) Hydra (9.0) Nmap (7.80) acumen-tlss tool acumen-tlsc tool acumen-sshs tool acumen-sshc tool
Bridge-1	Kali	2021.3	Bridge	TLS /SSH	Manually Set and Verified	tcpdump (4.9.3) acumen-MiTM tool
Bridge-2	Kali	2021.3	Bridge	TLS /SSH	Manually Set and Verified	tcpdump (4.9.3) acumen-MiTM tool

Name	OS	Version	Function	Protocols	Time	Tools (version)
Switch A	N/A	N/A	Switch	N/A	Manually Set and Verified	N/A
Switch B	N/A	N/A	Switch	N/A	Manually Set and Verified	N/A

5 Detailed Test Cases (TSS and the AGD Activities)

5.1 Mandatory Requirements

5.1.1 Security Audit (FAU)

5.1.1.1 FAU_GEN.1 Audit Data Generation

5.1.1.1.1 FAU_GEN.1 TSS

Objective:

- 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.
- For distributed TOEs the evaluator shall examine the TSS and 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 the 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 (as 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.

Evaluator Findings:

- The evaluator reviewed the TSS column titled **FAU_GEN.1** and ensured that it identifies the relevant key based on what information is logged. The TSS states:

  For the administrative task of generating/import of, changing, or deleting of cryptographic keys. The TOE uniquely identifies the relevant key depending on the type and format of the key:

  - **The TOE uses Distinguished Name for a key associated with an X.509 certificate,**
  - **The TOE uses the filename containing the public key and corresponding user account or client IP address for an SSH-based public key**

The evaluator reviewed the TSS and ensured 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 reviewed the TSS column titled **FAU_GEN.1** and ensured that the 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 (as applicable to the overall TOE). The evaluator confirmed that all components defined as generating audit information for a particular SFR contributed to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component covered all the SFRs that it implements. The TSS states:

  The TOE generates audit records for operation and administration of the Manager and Sensor. Administrative actions performed via Manager to manage the Manager or any Sensors are audited on Manager. Audit events are recorded in auditlog. Administrative actions performed via Sensor CLI (over SSH) to manage an individual Sensor are audited and cached by Sensor in a local file and then forwarded to the Manager.

- **Events logged in audit records include the items listed in Table 12, start-up and shut-down of the audit functions. The type of records generated for each component are determined by Table 11.**

Verdict:

PASS

5.1.1.1.2 FAU_GEN.1 AGD

Objective:

- The evaluator shall check the AGD 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).
- The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes.
- The evaluator shall examine the AGD 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 AGD satisfies the requirements related to it.

Evaluator Findings:

- The evaluator checked the AGD and ensured 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, was provided from the actual audit record). The section titled '**Appendix: Audit Log Records**' from the **Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** was used to determine the verdict of this assurance activity.
- The evaluator made a determination of the administrative actions related to TSF data related to configuration changes.
- The evaluator examined the AGD and made 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. Upon investigation, the evaluator found that **the following are applicable:**

Administrative Activity	**Method (Command/GUI Configuration)**	**Section**
Start-up and shut-down of the audit functions	**'syslog' command**	**'Manager Shell Commands' - IPS_11.1_Product_Guide**
Administrative login and logout	**Login : GUI (Manager) and CLI (Manager and Sensor)** **Logout : Logout button (Manager GUI) or 'exit' command (Manager and Sensor CLI)**	**Login : 'Accessing the Manager from a client machine' - IPS_11.1_Product_Guide; 'Reconfiguration of SSHD' - Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** **Logout : 'Menu bar', 'IPS CLI Commands - Normal Mode' and 'Manager Shell Commands' - IPS_11.1_Product_Guide**

Generating/import of, changing, or deleting of cryptographic keys	**Sensor: 'importsshpublickey' command** **Manager: 'publickeyAuth' command**	**Sensor: 'IPS CLI Commands - Normal Mode' and;** **Manager: 'Manager Shell Commands' - IPS_11.1_Product_Guide**
Resetting passwords	**Sensor: 'passwd' and 'userpasswd' commands** **Manager: GUI and 'set password' command**	**Sensor: 'IPS CLI Commands - Normal Mode' and;** **Manager: 'Management of users and user roles' (GUI) and 'Manager Shell Commands' (CLI) - IPS_11.1_Product_Guide**

- The evaluator documented the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator performed this activity as part of the activities associated with ensuring that the corresponding AGD satisfies the requirements related to it.

Verdict:

PASS.

5.1.1.1.3 FAU_GEN.1/IPS Audit Data Generation (IPS) TSS

Objective:

- The evaluator shall verify that the TSS describes how the TOE can be configured to log IPS data associated with applicable policies.
- The evaluator shall verify that the TSS describes what (similar) IPS event types the TOE will combine into a single audit record along with the conditions (e.g., thresholds and time periods) for so doing. The TSS shall also describe to what extent (if any) that may be configurable.
- For IPS_SBD_EXT.1, for each field, the evaluator shall verify that the TSS describes how the field is inspected and if logging is not applicable, any other mechanism such as counting that is deployed.

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that it describes how the TOE can be configured to log IPS data associated with applicable policies.
- The evaluator reviewed the TSS to ensure that it describes what (similar) IPS event types the TOE will combine into a single audit record along with the conditions (e.g., thresholds and time periods) for so doing. The TSS also describes to what extent (if any) that may be configurable.
- The evaluator reviewed the TSS to ensure that it describes how the field is inspected and if logging is not applicable, any other mechanism such as counting that is deployed.

The relevant information is found in the following section(s): TOE Summary Specification **FAU_GEN.1/IPS.**

Upon investigation, the evaluator found that the TSS states that: **The Sensor(s) generate audit records for the IPS events identified in Table. The Sensor(s) store the IPS audit data in the same file as general audit records for local events (e.g., trusted channel establishment, certificate validation errors). IPS audit records are configured through IPS policies. Data for multiple attacks is throttled into a single audit record when multiple instances of identical attacks (same attacker IP, target IP, and specific attack) are detected within a two-minute period. This threshold is also configurable via the alert suppression feature that the TOE provides.**

For information regarding logging for each field covered by IPS_SBD_EXT.1, refer the corresponding TSS[ST] section.

Furthermore, the TSS describes what IPS event types the TOE will combine into a single audit record along with the conditions for so doing. **The NSP Sensor(s) store the IPS audit data in the same file as general audit records for local events (e.g. trusted channel establishment, certificate validation errors).**

Verdict:

PASS.

5.1.1.1.4 FAU_GEN.1/IPS Audit Data Generation (IPS) AGD

Objective:

- The evaluator shall verify that the AGD describes how to configure the TOE to result in applicable IPS data logging.
- The evaluator shall verify that the AGD provides instructions for any configuration that may be done in regard to logging similar events (e.g., setting thresholds, defining time windows, etc.).

Evaluator Findings:

- The evaluator checked the AGD and ensured that it describes how to configure the TOE to result in applicable IPS data logging.
- The evaluator checked the AGD and ensured that it provides instructions for any configuration that may be done in regard to logging similar events (e.g., setting thresholds, defining time windows, etc.). The section titled **'Send Alert to Manage' under the 'Configure attack details' from the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that **the audit log options in the evaluated configuration can be configured using the mentioned steps and procedures. Furthermore, the evaluator determined that the operational guidance provides instructions for any configuration that may be done regarding logging similar events.**

Verdict:

PASS.

5.1.1.2 FAU_GEN.2 User Identity Association

5.1.1.2.1 TSS & AGD

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

5.1.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

5.1.1.3.1 FAU_STG_EXT.1 TSS

Objective:

- 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.
- 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.
- 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.
- 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 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.
- 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).
- 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.

Evaluator Findings:

- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured that it describes that the audit data are transferred to the external audit server over a secure TLS connection in realtime.
- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured it describes the amount of audit data that are stored locally; 50,000 records stored by the Manager. The TSS also describes what happens when the local audit data store is full; older audit records are deleted. The TSS also describes how these records are protected against unauthorized access. The TSS states:

  There is no filesystem access to any administrative users (as the CLI is provided by a zebra shell with a limited set of commands).
- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured that it describes that the TOE is a distributed TOE that contains TOE components (Sensor) that cannot store audit data locally on themselves but need to transfer audit data to other TOE components (Manager) that can store audit data locally.
- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured that for distributed TOEs it contains a list of TOE components (Manager) that store audit data locally.
- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured that for distributed TOEs that contain components which do not store audit data locally (Sensor) but transmit their generated audit data to other components (Manager) it contains a mapping between the transmitting and storing TOE components.
- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured that it details the behaviour of the TOE when the storage space for audit data is full. The option 'overwrite previous audit record' is selected and the TSS provides description "The most recent 50,000 audit records are retained on the Manager; older audit records are deleted".
- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured that it details that the transmission of audit information to an external IT entity is done in real- time.

- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured that 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 component 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 TSS states:

  The TOE is a distributed TOE and includes a Manager and one or more Sensors. The TOE stores audit data locally on the Manager. The Sensors have limited local storage and hence they send audit files to the Manager for storage. These files are marked to denote which Sensors they are received from.
- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured that 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 describes the behaviour when local storage space or buffer space is exhausted. The TSS states:
  - **By default, the manager stores 50,000 audit records in a local database. Manager also forwards all audit records to a syslog server over a TLS secured connection in real-time.**
  - **When the default threshold of 50,000 audit records is met, the most recent 50,000 audit records are retained on the Manager and the older audit records are overwritten by the newer ones.**
  - **The Sensor generates an auditlog file that is no more than 128MB. The file is purged from the disk when the audit log is uploaded to the Manager and a new auditlog file is started with a start marker.**

Verdict:

PASS.

5.1.1.3.2 FAU_STG_EXT.1 AGD

Objective:

- The evaluator shall also examine the AGD 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.
- The evaluator shall also examine the AGD 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.
- The evaluator shall also ensure that the AGD 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.

Evaluator Findings:

- The evaluator examined the AGD section titled '**Forward alert notifications from the Manager to a syslog server' of the IPS_11.1_Product_Guide** and ensured 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.
- The evaluator also examined the AGD section titled '**Forward alert notifications from the Manager to a syslog server' of the IPS_11.1_Product_Guide** and determined that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. The AGD section titled **'Alert notification options' of IPS_11.1_Product_Guide states, The Manager forwards all the audit records to a syslog server over a TLS secured connection in real-time.**
- The evaluator also found that **the TOE does not require configuration for log overwriting since end users lack shell access and cannot modify these mechanisms. The AGD section titled 'Manager informational faults'**

IPS_11.1_Product_Guide provides information about the default values for FAU_STG_EXT.1.3 and the resulting behavior of the TOE. The description corresponds to those described in the TSS.

Verdict:

PASS.

5.1.1.4 FAU_STG_EXT.4 Protected Local audit event storage for distributed TOEs & FAU_STG_EXT.5 Protected Remote audit event storage for Distributed TOEs

5.1.1.4.1 FAU_STG_EXT.4 & FAU_STG_EXT.5 TSS

Objective:

- The evaluator examines the TSS to confirm that it describes which TOE components store their security audit events locally and which send their security audit events to other TOE components for local storage. For the latter, the target TOE component(s) which store security audit events for other TOE components shall be identified. For every sending TOE component, the corresponding receiving TOE component(s) need to be identified. For every transfer of audit information between TOE components it shall be described how the data is secured during transfer according to FTP_ITC.1 or FPT_ITT.1.
- For each TOE component which does not store audit events locally by itself, the evaluator confirms that the TSS describes how the audit information is buffered before sending to another TOE component for local storage.

Evaluator Findings:

- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured that it describes that the Manager TOE component store their security audit events locally and Sensor TOE component sends their security audit events to the Manager for local storage. For every transfer of audit information between TOE components the TSS described that the data is secured during transfer using TLS according to FPT_ITT.1.
- The evaluator examined the TSS column titled **FAU_STG_EXT.1** and ensured that the TSS describes how the audit information is buffered before sending to another TOE component for local storage. The TSS states:

  The Sensor generates an auditlog file that is no more than 128MB. The file is purged from the disk when the audit log is uploaded to the Manager and a new auditlog file is started with a start marker. When the file is exhausted, new events are dropped and a sysevent is sent to the Manager informing no further audit messages will be recorded until the log is purged.

Verdict:

PASS.

5.1.1.4.2 FAU_STG_EXT.4 & FAU_STG_EXT.5 AGD

Objective:

- The evaluator shall examine the guidance documentation to ensure that it describes how the link between different TOE components is established if audit data is exchanged between TOE components for local storage. The guidance documentation shall describe all possible configuration options for local storage of audit data and provide all instructions how to perform the related configuration of the TOE components.
- The evaluator shall also ensure that the guidance documentation describes for every TOE component which does not store audit information locally how audit information is buffered before transmission to other TOE components.

Evaluator Findings:

- The evaluator examined the AGD section titled '**Syslog notifications', 'Sensor functionality' and 'Alert notification options' of the IPS_11.1_Product_Guide** and ensured that it describes how the link between different TOE components is established if audit data is exchanged between TOE components for local storage.

  By default, the Sensor forwards alert information to the Manager. All the manager-sensor communications happen over TLS, and if configured, the Manager forwards all the audit records to a syslog server over a TLS secured connection in real-time.

  The evaluator also examined the AGD section titled '**Syslog notifications' of the IPS_11.1_Product_Guide** and ensured it describes all possible configuration options for local storage of audit data and found that there is no needs for configuration of the TOE components, Because

  'By default, the Sensor forwards alert information to the Manager'.

- The evaluator examined the AGD section titled '**Sensor functionality' of the IPS_11.1_Product_Guide** and ensured that it describes for every TOE component which does not store audit information locally how audit information is buffered before transmission to other TOE components.

  The Sensor generates an auditlog file that is no more than 128MB. The file is purged from the disk when the audit log is uploaded to the Manager and a new auditlog file is started with a start marker.

  All the manager-sensor communications happen over TLS.

Verdict:

PASS.

5.1.2 Cryptographic Support (FCS)

5.1.2.1 FCS_CKM.1 Cryptographic Key Generation

5.1.2.1.1 FCS_CKM.1 TSS

Objective:

- 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.
- If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

Evaluator Findings:

- The evaluator ensured that the TSS column titled **FCS_CKM.1** identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator examined the TSS to verify that it identifies the usage for each scheme.
- The evaluator examined the TSS column titled **FCS_CKM.1** and ensured that if the ST specifies more than one scheme, it identifies the usage for each scheme. The TSS states:

  The TOE generates 2048-bit RSA keys as specified in FIPS Pub 186-4. These keys are available for mutual identification of the TOE components in an FPT_ITT.1 Intra-TSF Trusted Channel using TLS with mutual authentication.
  RSA keys are used to identify the Manager (Web GUI) to the administrator. The Manager also uses RSA keys to identify the remote syslog server. Both of these communications use TLS without mutual authentication. The RSA-based keys are generated and only used for identification and authentication purposes (including digital signatures). They are not used for key exchange.

The TOE generates P-256 and P-384 curve ECDSA keys as specified in FIPS Pub 186-4. When using SSH, the Manager and Sensor uses an ECDSA key with P-256 curves to identify itself to the administrators. Similarly, a Sensor uses an ECDSA key with P-256 curves to identify itself to the update server (SCP server).
The Manager and Sensor uses a 2048-bit RSA key, or ECDSA key with P-256 to authenticate an administrator that is using public-key based authentication mechanism.

Verdict:

PASS.

5.1.2.1.2 FCS_CKM.1 AGD

Objective:

The evaluator shall verify that the AGD 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.

Evaluator Findings:

The evaluator verified that the AGD sections titled **'Sensor as the SSH client' and 'Sensor as the SSH server' under 'Sensor CLI for Certification' provides information related to generation of RSA and ECDSA based public keys and 'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** provides information about to TLS and related key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target. The TLS channels are pre-configured to use the selected key generation scheme(s) and key size(s) defined in the Security Target.

Verdict:

PASS.

5.1.2.2 FCS_CKM.2 Cryptographic Key Establishment

5.1.2.2.1 FCS_CKM.2 TSS

Objective:

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

Evaluator Findings:

- The evaluator ensured that the TSS column titled **FCS_CKM.2** identifies the key establishment schemes supported by the TOE. If the ST specifies more than one scheme, the evaluator examined the TSS to verify that it identifies the usage for each scheme. The TSS states:

  The TOE uses ECDHE ciphers for key exchange with TLS. The ECDHE keys that are generated uses P-256, and P-384 curves and are generated as specified in FIPS Pub 186-4.

Table 4 describes the key establishment schemes and how they are used by the TOE.

Scheme	SFR	Service
ECDHE	FCS_TLSC_EXT.2 FPT_ITT.1	ECDHE Intra-TSF Trusted Channel

Scheme	SFR	Service
ECDHE	FCS_TLSC_EXT.1	Syslog Server
ECDHE	FCS_TLSS_EXT.1	Administration
ECDHE	FCS_TLSS_EXT.2	ECDHE Intra-TSF Trusted Channel
ECDH	FCS_SSHS_EXT.1 FCS_SSHC_EXT.1	Administration Importing TOE updates

Table 4 Key Establishment Schemes

The TOE acts as a sender and a recipient when performing Elliptic Curve Diffie-Hellman. ECDHE key establishment is performed as specified in SP 800-56A Revision 3.

- The evaluator examined the TSS column titled **FCS_CKM.2** and ensured that if the ST specifies more than one scheme, it identifies the usage for each scheme.

  DH-14 is not claimed; hence the activity is not applicable.

Verdict:

PASS.

5.1.2.2.2 FCS_CKM.2 AGD

Objective:

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

Evaluator Findings:

The evaluator verified that the AGD section titled **'Sensor as the SSH client' under 'Configuration of Trellix IPS for Certification' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** instructs the administrator how to configure the TOE to use the selected key establishment scheme(s) for all cryptographic protocols defined in the Security Target.

Verdict:

PASS.

5.1.2.3 FCS_CKM.4 Cryptographic Key Destruction

5.1.2.3.1 FCS_CKM.4 TSS

Objective:

- 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 for2). 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.

- 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).
- 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.
- 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.
- 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.

Evaluator Findings:

- The evaluator ensured that the TSS column titled **FCS_CKM.4** identifies the key destruction mechanism supported by the TOE.
- The evaluator examined the TSS column titled **FCS_CKM.4** and ensured the TOE meets all requirements specified in FIPS 140-2 for destruction of keys and Critical Security Parameters (CSPs).
- The evaluator examined the TSS to ensure that it lists each type of plaintext key material and its origin and storage location. The TSS entry for FCS_CKM.4 in the section 6 titled TOE Security Functional Requirements as well as Key Zeroization of ST was used to determine the verdict of this assurance activity.

  According to the TSS the following plaintext keys are stored in their respective memory location:

  For manager(s):

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
Manager Public/private keys (persistent)	Generated using java keytool. RSA 2048 bit key	Stored in DB, in plaintext, protected with passphrase	On deinstallation of Manager with deletion of DB
Sensor Secret Key	Generated using custom hashing mechanism Mutual authentication parameter for the Manager and Sensor during joining	Stored in DB, in plaintext, protected with passphrase	On deletion of Sensor Entry from Manager
SSH Host Public/Private Key	ECDSA P-256 curve key used to authenticate MLOS Appliance to remote client during SSH	Plaintext, key is stored in hex form in a file.	Delete public/private key from system, remove the SSH user entry from known hosts file
SSH Session Key	Session keys used with SSH, AES 128/256, ECDH Private Key P-256	Plaintext session keys stored in RAM used for SSH session agreement	Zeroized in RAM on reboot using OpenSSL scrubbing Method

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
TLS Session Key	Session keys used with TLS, AES 128/256, HMAC-SHA-256/384, ECDH P-256, P-384	Plaintext session keys stored in RAM used for TLS session agreement	Memory scrubbed using OpenSSL method upon termination of session
User password	User generated, Stored PBKDF2 with HMAC-SHA-512 in the DB	Plaintext value held in RAM as entered by user.	On deinstallation of Manager with deletion of DB
Block Cipher (CTR) DRBG State	To generate random bits needed for asymmetric key, symmetric key, nonce, and salt generation	Plaintext seed key and state of RNG held in RAM	Memory scrubbed by OpenSSL once seed passed to RNG. RNG scrubbed using OpenSSL method during normal shutdown.
Trellix Manager Image Verification Key	RSA 2048 bit key used to authenticate IPS Manager firmware images	Plaintext, Loaded to the disk and into RAM	N/A – Public Key

For sensor(s):

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
1	Administrator Passwords	Authentication of the “admin” role through console and SSH login. Extended services are given to the “admin” role by using the “support” and “private” passwords. This extended service of “support” and “private” are configurable via CLI (privatemode enable|disable)	N/A. Default “admin” password set at manufacturing time and is then set by the Admin. The “support” and “private” passwords are set in image files and can only be changed in new image file.	The “admin” password is stored via HMAC-SHA-512 hash in Linux shadow file. The “support” and “private” passwords are stored via HMAC-SHA-512 hash in shell.conf file.	Entry: During login and when being set through a CLI command. Also, via enable command in CLI to allow extended services. Output: Never output	No Zeroize Service Reqd.

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
		and are enabled by default.				
2	Sensor User Passwords (Users created by admin using "adduser" CLI)	Authentication of "user" accounts through console and SSH login. Extended services are given to the "user" accounts by using the "support" or "private" passwords. This extended service of "support" and "private" are only for users with "admin" access and are configurable via CLI (privatemode enable|disable) and are enabled by default.	N/A. Externally generated.	The "user" password is stored via HMAC-SHA-512 hash in Linux shadow file. The treatment is like "admin" except these are not pre-generated with defaults within the image. The "support" and "private" passwords are stored via HMAC-SHA-512 hash in shell.conf file.	Entry: During login and when being set through a CLI command. Also, via enable command in CLI to allow extended services. Output: Never output	No Zeroize Service Reqd.
3	3rd Party SNMP Client Privacy and Authentication Keys	Authentication of the 3rd Party SNMP role.	N/A. Externally generated.	Encrypted on Storage Media (internal SSD) and temporarily stored in RAM as plaintext.	Entry: Initially set RSA key wrapped by Manager. Also entered during authentication. Output: Never output	Zeroized from Storage Media (internal SSD) on resetconfig and internal rescue. Zeroized from RAM on each reboot.
4	Manager SNMP Client Privacy and Authentication Keys	Authentication of the Manager SNMP role.	N/A. Externally generated.	Only stored temporarily in RAM as plain text.	Entry: Received from Manager as plain text key through TLS channel.	Zeroized from RAM on each reboot.

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
					Output: Only displayed in the private mode command.	
5	Manager Initialization Secret (i.e., Manager "Shared Secret")	Mutual authentication parameter for the sensor and Manager during initialization. Note: The Manual Key Entry Test is required. This is already being done by forcing the user to enter the key twice.	N/A. Externally generated.	Temporarily in Plaintext in RAM	Entry: Entered by the User through CLI, 'set sharedsecretkey' command Output: Never output	Zeroized after reboot.
6	Proprietary File Transfer Channel Session Key (Secrete and IV are encrypted by Manager using the Sensor public key)	Used to encrypt data packages across the Proprietary File transfer channel	N/A. Entered through SNMPv3 channel for Proprietary File transfer session. Keys are provided through SNMP by encrypting using sensor public key.	Plaintext in RAM.	Entry: RSA key wrapped Output: Never output	Zeroized in RAM on reboot
7	SSH Host Private Keys (ssh_host_ECDSA_key) (Sensor as SSH Server)	Authentication of sensor to remote terminal for CLI access	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Plaintext in Storage Media (internal SSD) and temporarily in RAM	Entry: Never entered Output: Never output	Zeroized from Storage Media (internal SSD) on resetconfig or internal Rescue, Zeroized from RAM on reboot

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
8	SSH Session Private Keys (Sensor as SSH Server)	Set of ephemeral EC Diffie-Hellman P-256, AES 128/256 bit, and HMAC (SHA-256/512) keys created for each SSH session.	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Temporarily in Plaintext in RAM	Entry: Never entered Output: Never output	Zeroized in RAM on reboot and zeroized by openSSH library upon every SSH session closure.
9	SSH Client Private Keys (id_ecdsa) (Sensor as SSH Client)	Authentication of sensor to remote server for SCP communication.	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Encrypted using a different set of RSA private/public key pair and stored in Storage Media (internal SSD)	Entry: Never entered Output: Never output	Zeroized from Storage Media (internal SSD) on resetconfig or internal Rescue.
10	SSH Session private Keys (Sensor as SSH Client)	Set of ephemeral EC Diffie-Hellman P-256, AES 128/256 bit, and HMAC (SHA-256/512) keys created for each SCP session.	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Temporarily in Plaintext in RAM	Entry: Never entered Output: Never output	Zeroized in RAM on reboot and zeroized by openSSH library upon every SCP session closure.
11	TLS Sensor Private Key (alert/sysEvent channel for Manager) (skeyman)	RSA 2048-bit key used for authentication of the sensor to Manager.	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Plaintext in EEPROM and temporarily in RAM	Entry: Never entered Output: Never output	Zeroized from EEPROM on resetconfig or internal Rescue, Zeroized from RAM on reboot
12	TLS Session private Keys (for Manager)	Set of ephemeral EC Diffie Hellman P-256, AES 128/256 bit and HMAC (SHA-256/512 bit) keys created for each TLS session with the Manager.	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Temporarily in Plaintext in RAM	Entry: Never entered Output: Never output	closeAndCleanUpEMSConn in emsconnection.c cleans up all application contexts and orphans any

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
						objects (keys) in OpenSSL. This occurs in lieu of a module reboot to clear RAM. Also, zeroized on De-install and Reboot.
13	Seed for RNG	Seed created by NDRNG and used to seed the Block Cipher (CTR) DRBG. The Nonce is 128 bits and the Entropy Input is 256 bits for a total seed size of 384 bits.	Internally using the NDRNG, which is based on CPU jitter (time delta) value.	NA	Entry: Never entered Output: Never output	zeroized as part of openSSL scrubbing and in RAM on reboot
14	DRBG Internal State	V and Key used by the DRBG to generate pseudo-random numbers	Internally using the NDRNG, which is based on CPU jitter (time delta) value.	Plaintext temporarily in RAM	Entry: Never entered Output: Never output	Zeroized as part of openSSL scrubbing and in RAM on reboot
15	Entropy Input String	8192-bit output string from the Jitter Entropy library	Output from the NDRNG	Plaintext temporarily in RAM	Entry: Never entered Output: Never output	Zeroized as part of openSSL scrubbing and in RAM on reboot
16	Trellix FW Verification Key	2048-bit RSA/SHA-256 public key used to authenticate software images loaded into the module	Generated in Trellix secure lab and embedded inside image	Plaintext on boot media	Entry: Never entered Output: Never output	The key can be changed only by change in new image file

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
17	SSH Host Public Key (Sensor as Server)	ECDSA P-256-bit key used to authenticate the sensor to the remote client during SSH.	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Plaintext on internal Storage Media (SSD) and in RAM as plaintext.	Entry: Never entered Output: During SSH handshake.	Zeroized from Storage Media (internal SSD) on resetconfig or internal Rescue, Zeroized from RAM on reboot
18	SSH Remote Client Public Key (Sensor as Server)	ECDSA and RSA P-256-bit key used to authenticate the remote client to the sensor during SSH.	Externally generated	Plaintext in RAM and Storage Media (SSD)	Entry: During SSH handshake. Output: Never output	Zeroized from Storage Media (internal SSD) on resetconfig or internal Rescue, Zeroized from RAM on reboot
19	SSH Session Public Key (Sensor as Server)	EC Diffie-Hellman P-256-bit session key created for each SSH session	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Temporarily in RAM stored as Plaintext	Entry: Never entered Output: Never output	Zeroized in RAM on reboot and zeroized by openSSH library upon every SSH session closure.
20	SSH Client Public Key (Sensor as Client)	ECDSA P-256-bit key used to authenticate the sensor to the remote server during SCP.	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Plaintext on internal Storage Media (SSD) and in RAM as plaintext.	Entry: Never entered Output: During SCP handshake.	Zeroized from Storage Media (internal SSD) on resetconfig or internal Rescue, Zeroized from RAM on reboot

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
21	SSH Session Public Key (Sensor as Client)	EC Diffie-Hellman P-256-bit session key created for each SCP session	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Temporarily in RAM stored as Plaintext	Entry: Never entered Output: Never output	Zeroized in RAM on reboot and zeroized by openSSH library upon every SCP session closure.
22	TLS Sensor Public Key (for Manager)	RSA 2048-bit key used to authenticate the sensor to Manager during TLS connections.	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Plaintext on internal Storage Media (SSD) and in RAM	Entry: Never entered Output: During initial TLS handshake	Zeroized during resetconfig/internal rescue and deinstall
23	TLS Manager Public Key	RSA 2048-bit key used to authenticate Manager to sensor during TLS connections.	Externally generated.	Plaintext on internal Storage Media (SSD) and in RAM	Entry: During initial TLS handshake Output: Never output	Zeroized during resetconfig/internal rescue and deinstall
24	TLS Session Public Key	EC Diffie-Hellman P-256 bit session key created for each TLS session	Internally using the Block Cipher (CTR) DRBG provided by OpenSSL	Temporarily in RAM stored as Plaintext	Entry: Never entered Output: Never output	closeAndCleanUpEMSConn in emsconnection.c cleans up all application contexts and orphans any objects (keys) in OpenSSL. This occurs in lieu of a module reboot to clear RAM. Also, zeroized on De-install and Reboot.

- **The TOE does not use of a value that does not contain any CSP to overwrite keys, hence, not relevant.**

Verdict:

PASS.

5.1.2.3.2 FCS_CKM.4 AGD

Objective:

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 command3 and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

Evaluator Findings:

Upon investigation, the evaluator found **that the TSF performs all the key destruction mechanism as specified in FIPS 140-2 for destruction of keys and Critical Security Parameters (CSPs). The evaluator reviewed the AGD documentation for the TOE and found no situation that would prevent or delay key destruction.**

Also section '**Key destruction (zeroization) mechanism details and exceptions**' of Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide states; **There are no exceptions to the Sensor and Manager. For more details, refer to the tables Manager Key zeroisation and Sensor Key zeroisation under the topic [Cryptographic Key Destruction] within the [Trellix IPS Sensor and Manager Appliances version 11.1 Security Target] document.**

Verdict:

PASS.

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

5.1.2.4.1 FCS_COP.1/DataEncryption TSS

Objective:

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.

Evaluator Findings:

The evaluator examined the TSS column titled **FCS_COP.1/DataEncryption** identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption. The TSS states:

The TOE performs AES 128- and 256-bit encryption in GCM mode to secure TLS and SSH communication channels.

Verdict:

PASS.

5.1.2.4.2 FCS_COP.1/DataEncryption AGD

Objective:

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.

Evaluator Findings:

Upon investigation, the evaluator found that **the TOE does not require configuration for key size(s) and mode(s) for data encryption/decryption, since it is pre-configured and fixed. The information about the default configuration can be found in the sections 'Sensor features in FIPS compliant images' and 'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide and the section 'Reconfiguration of SSH and SSHD' of the Trellix Intrusion Prevention System Manager Appliance Product Guide.**

Verdict:

PASS.

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

5.1.2.5.1 FCS_COP.1/SigGen TSS

Objective:

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

Evaluator Findings:

The evaluator ensured that the TSS column titled **FCS_COP.1/SigGen** identifies the cryptographic algorithm and key size supported by the TOE for signature services.

Upon investigation, the evaluator found that the TSS states that: **The TOE performs RSA 2048-bit SigGen to support TLS functions. The TOE performs RSA 2048 bit SigVer to support TLS, X.509, and trusted update functions. The TOE performs RSA 2048 bit and ECDSA 256-bit SigVer to support administrative authentication while using public-key mechanism. The TOE performs ECDSA 256-bit SigGen and SigVer to only support the SSH public key-based authentication functions (host key).**

Verdict:

PASS.

5.1.2.5.2 FCS_COP.1/SigGen AGD

Objective:

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.

Evaluator Findings:

Upon investigation, the evaluator found that **the TOE does not require configuration for using the cryptographic algorithm and key size for signature services, since it is pre-configured and fixed. The information about the default configuration can be found in the sections 'Sensor features in FIPS compliant images' and 'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide and the section 'Reconfiguration of SSH and SSHD' of the Trellix Intrusion Prevention System Manager Appliance Product Guide.**

Verdict:

PASS.

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

5.1.2.6.1 FCS_COP.1/Hash TSS

Objective:

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.

Evaluator Findings:

The evaluator ensured that the TSS column titled **FCS_COP.1/Hash** identifies the hash functions associated with other TSF cryptographic functions.

Upon investigation, the evaluator found that the TSS evaluator found that the TSS states that: **The TOE performs SHA-1, SHA-256, SHA-384, and SHA-512 hashing. These hashes are used for SigGen and SigVer operations. SHA-1, SHA-256 and SHA-512 are used in the SSH, while SHA-256 and SHA-384 are used for the TLS functionalities. The hash algorithms are also used in the associated HMAC algorithms.**

Verdict:

PASS.

5.1.2.6.2 FCS_COP.1/Hash AGD

Objective:

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

Evaluator Findings:

Upon investigation, the evaluator found that **the TOE does not require configuration for hash sizes, since it is pre-configured and fixed, and these mechanisms cannot be modified. The information about the default configuration for the hash sizes for different protocols can be found in the sections 'Sensor features in FIPS compliant images' and 'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide.**

Verdict:

PASS.

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

5.1.2.7.1 FCS_COP.1/KeyedHash TSS

Objective:

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.

Evaluator Findings:

The evaluator ensured that the TSS column titled **FCS_COP.1/KeyedHash** identifies the HMAC specifications being used by the TOE.

Upon investigation, the evaluator found that the TSS states that: **The TOE uses HMAC-SHA-256 for TLS KDF and TLS message authentication. HMAC-SHA-256 uses a 256 bit key, 512 bit block size, and 256 bit message digest size.**

The TOE uses HMAC-SHA-384 for TLS KDF and TLS message authentication on the Manager. HMAC-SHA-384 uses a 384 bit key, 1024 bit block size, and 384 bit message digest size.

The TOE uses HMAC-SHA-512 in PBKDF2 for password obfuscation. HMAC-SHA-512 uses a 512 bit key, 1024 bit block size, and 512 bit message digest size.

The TOE uses 'implicit' keyed-hash message authentication for the SSH client and server functionalities, which make use of AES-GCM ciphers capable of providing integrity on their own.

Verdict:

PASS.

5.1.2.7.2 FCS_COP.1/KeyedHash AGD

Objective:

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.

Evaluator Findings:

Upon investigation, the evaluator found that **the TOE does not require configuration for the values used by the HMAC function. The information about the default configuration can be found in the section 'Sensor features in FIPS compliant images' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide. This section provides information regarding algorithms used during SSH.**
The section 'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide provides information related to the algorithms used during TLS.

Verdict:

PASS.

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

5.1.2.8.1 FCS_RBG_EXT.1 TSS

Objective:

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.

Evaluator Findings:

The evaluator examined the TSS column titled **FCS_RBG_EXT.1** and determined that it specifies the DRBG type **(Counter DRBG)**, identifies the entropy source(s) seeding the DRBG **(hardware-based noise source)**, and state the assumed or calculated min-entropy supplied either separately by each source or the min- entropy contained in the combined seed value **(The 1024 bytes of data contain at least 256-bit of entropy)**.

Verdict:

PASS.

5.1.2.8.2 FCS_RBG_EXT.1 AGD

Objective:

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

Evaluator Findings:

The evaluator confirmed that the AGD contains appropriate instructions for configuring the RNG functionality. Upon investigation, the evaluator found that no configuration is required for implementation of the RNG functionality. **The section titled 'Device bootup processing' provides information about the default configurations**.

Verdict:

PASS.

5.1.3 Identification and Authentication (FIA)

5.1.3.1 FIA_AFL.1 Authentication Failure Management

5.1.3.1.1 FIA_AFL.1 TSS

Objective:

- 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 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).

Evaluator Findings:

- The evaluator examined the TSS column titled **FIA_AFL.1** and determined 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 also describes 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 TSS States:

  All management of the TOE is performed though the Web UI of Manager component, and CLI of individual components (Manager and Sensor). Identification and authentication are required for both local and remote administrator access. Remote access to the TOE is via an SSH (provides CLI access) or HTTPS session (provides Web Ui access) from the Management Workstation. Local access to the TOE is via the appliance console port (provides CLI access).

  The TOE also tracks the number of sequential failed authentication attempts for each user. Upon exceeding the configured threshold value, with the default being 3 failures, the TOE locks the account until admin unlocks the user using the CLI command on sensor. On the Manager, such an account remains locked until the configured lockout period elapses. During this time, entering the correct password for the locked account will still result in an authentication failure. The TSF also allows a local administrator to clear the lock. The local administrator cannot be locked out. Any successful authentication resets the counter to zero.
- The evaluator examined the TSS column titled **FIA_AFL.1** and confirmed 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). The TSS states: **The local administrator cannot be locked out.**

Verdict:

PASS.

5.1.3.1.2 FIA_AFL.1 AGD

Objective:

- The evaluator shall examine the AGD 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.

- The evaluator shall examine the AGD 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.

Evaluator Findings:

- The evaluator examined the AGD sections titled **'Set up account lockout parameters', 'Customize the unlock time and maximum authentication attempts' and the 'unlockuser WORD' under 'CLI commands' of the IPS_11.1_Product_Guide** and ensured that instructions for configuring the number of successive unsuccessful authentication attempts **can be configured. The section 'Customize the unlock time and maximum authentication attempts' provides information on how a locked out remote user can once again successfully log on to the manager. The manager's locking mechanism is time-driven. The command 'unlockuser WORD' under 'CLI commands' mentions the action to be taken for unlocking a locked user on a sensor device.**
- The evaluator examined the AGD sections titled **'Set up account lockout parameters' and the 'unlockuser WORD' under 'CLI commands' of the IPS_11.1_Product_Guide** and confirmed that it describes, and identifies the importance of, any actions that are required and ensured 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.

Verdict:

PASS.

5.1.3.2 FIA_PMG_EXT.1 Password Management

*5.1.3.2.1 FIA_PMG_EXT.1 TSS **[TD0792 applied]***

Objective:

- The evaluator shall check that the TSS lists the supported special character(s) for the composition of administrator passwords.
- The evaluator shall check the TSS to ensure that the minimum_password_length parameter is configurable by a Security Administrator.
- 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.

Evaluator Findings:

- The evaluator examined the TSS and determined that it contains the lists of the supported special character(s) and minimum and maximum number of charters supported for administrator passwords.

The relevant information is found in the following section(s): TOE Summary Specification **FIA_PMG_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **Authentication of an administrator is configured to be through use of a username/password. Following are the enforced password complexity requirements for the Sensor CLI:**

- **Minimum length of 15 characters.**
- **Contains at least 2 lower case, 2 upper case, 2 numeric and 2 special characters ("!", "@", "#", "$", "%", "^", "&", "*", "(", ")").**

The enforced Manager CLI password requirements are as follows:

- **Minimum length of 15 characters.**

- **Contains at least 1 lowercase, 1 upper case, 1 numeric and 1 special character (“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”).**

The minimum password length is configurable by an administrator in a range of 8 to 64 characters along with its character composition for the Manager GUI. However, the following settings are recommended for a CC configuration:

- **Minimum length of 15 characters.**
- **Contains at least 2 lowercase, 2 upper case, 2 numeric and 2 special characters (“~”, “`”, “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “_”, “+”, “-“, “=”, “[“, “]”, “{“, “}”, “\”, “|”, “;”, “:” “‘”, “‘”, “,”, “.”, “<”, “>”, “?” and “/”).**

Verdict:

PASS.

5.1.3.2.2 FIA_PMG_EXT.1 AGD

Objective:

The evaluator shall examine the AGD to determine that it:

a) identifies the characters that may be used in passwords and provides the AGD 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.

Evaluator Findings:

The evaluator examined the AGD sections titled **‘Handling user password between FIPS and non-FIPS Sensor images’ and ‘Password requirements for Manager CLI and GUI’ of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide, which describe the password requirements for the Sensor and Manager CLI.**

The evaluator also examined the sections **‘Add users’** and **‘Configure password complexity settings’** of the **IPS_11.1_Product_Guide which describe the password requirements for Manager GUI** and determined that it:

a) identifies the characters that may be used in passwords and provides the AGD 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.

Verdict:

PASS.

5.1.3.3 FIA_UIA_EXT.1 User Identification and Authentication

5.1.3.3.1 FIA_UIA_EXT.1 TSS

Objective:

- 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”.

- 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.
- 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.
- 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.

Evaluator Findings:

- The evaluator examined the TSS column titled **FIA_UIA_EXT.1** and determined that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description contains information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon". The TSS States:

  During entry of the password, each character entered is masked with a "*" on the Manager GUI prompt and with ""(blank space) on the Manager and sensor SSH prompts when progress is reflected on the screen. If an authentication attempt fails (either the username is not recognized or the password is incorrect), the same "Login failed" error message is presented.

  If successful, the user's session is initiated under the assigned role. If unsuccessful, the authentication attempt fails and the connection is immediately terminated.

- The evaluator examined the TSS column titled **FIA_UIA_EXT.1** and determined that it describes which actions are allowed before user identification and authentication. The description covers authentication and identification for local and remote TOE administration. The TSS States:

  Prior to logon via console, SSH, and web GUI, a consent banner is displayed to the administrator warning that proceeding with authentication is consenting to the terms of use of the TOE.

- The evaluator examined the TSS column titled **FIA_UIA_EXT.1** and ensured that it 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 describes 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 TSS States:

  All management of the TOE is performed though the Web UI of Manager component, and CLI of individual components (Manager and Sensor). Identification and authentication are required for both local and remote administrator access.

- The evaluator examined the TSS column titled **FIA_UIA_EXT.1** and ensured that it describes for each TOE component which actions are allowed before user identification and authentication. The description covers 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 describes any unauthenticated services/services that are supported by the component. The TSS States:

  Prior to logon via console, SSH, and web GUI, a consent banner is displayed to the administrator warning that proceeding with authentication is consenting to the terms of use of the TOE. This warning banner is displayed on all management interfaces for all TOE components.

Verdict:

PASS.

5.1.3.3.2 FIA_UIA_EXT.1 AGD

Objective:

The evaluator shall examine the AGD 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 login method, the evaluator shall ensure the AGD 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 AGD provides sufficient instruction on limiting the allowed services.

Evaluator Findings:

The evaluator examined the AGD section titled **'Accessing the Manager from a client machine' and section 'Logging onto the Sensor via an SSH client' of the IPS_11.1_Product_Guide provides information for logging into the manager via WEB GUI and sensor via SSH respectively.**

The section 'Connect the cable to the Console port' of the Trellix Intrusion Prevention System NS-series Sensor Product Guide provides information for logging into the sensor via console. The console ports highlighted in sections 'Manager Appliance front panel description', 'Manager Appliance back panel description' and section 'SSH public key based authentication for Manager Appliance' of the Trellix Intrusion Prevention System Manager Appliance Product Guide provides information for logging onto the manager via console and SSH respectively.

The section titled 'SSH public key based authentication for Sensor' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide provides information related to public key based authentication for the sensor and determined 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 login method, the evaluator ensured that the AGD provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator determined that the AGD provides sufficient instruction on limiting the allowed services.

Verdict:

PASS.

5.1.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

Objective:

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.

Evaluator Findings:

Evaluation Activities for this requirement are covered under those for FIA_UIA_EXT.1. No other authentication mechanisms are specified**.**

Verdict:

PASS.

5.1.3.5 FIA_UAU.7 Protected Authentication Feedback

5.1.3.5.1 FIA_UAU.7 AGD

Objective:

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

Evaluator Findings:

The evaluator examined the AGD and determined that any necessary preparatory steps to ensure authentication data is not revealed while entering for each local login allowed. Upon investigation, the evaluator found that no configuration is required to maintain the confidentiality of the authentication data. The entirety of AGD was used for this activity.

Verdict:

PASS.

5.1.4 Security Management (FMT)

5.1.4.1 FMT_MOF.1/ManualUpdate

5.1.4.1.1 FMT_MOF.1/ManualUpdate AGD

Objective:

- The evaluator shall examine the AGD to determine that any necessary steps to perform manual update are described. The AGD shall also provide warnings regarding functions that may cease to operate during the update (if applicable).
- For distributed TOEs the AGD 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 AGD shall also provide warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update (if applicable).

Evaluator Findings:

- The evaluator examined the AGD sections titled **'Device software' under 'Trellix IPS Protection Status', 'Upgrade' under 'Manager Shell Commands', 'loadimage' and 'loadsavedimage' under 'IPS Commands - Normal Mode' of the Trellix IPS 11.1.x Product Guide and the section 'Upgrade Paths' of the Trellix IPS 11.1 FIPS and CC Certification Guide** and determined that any necessary steps to perform manual update are described.

  The AGD also provides warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update.
- The evaluator examined the AGD sections titled **'Device software' under 'Trellix IPS Protection Status' where information regarding downloading of sensor software can be found. The section 'Upgrade' under 'Manager Shell Commands' describes the process for upgrading the manager software using manager CLI. The sections 'loadimage', 'loadsavedimage' and 'loadsavedimagefrompeer' under 'IPS Commands - Normal Mode' describe the process for upgrading the sensor software using the sensor CLI.**

  The evaluator ensured that the AGD describes all steps how to update all TOE components. **The order is not relevant to the update process. Both sensor and manager can be upgraded independently in any order.**

Verdict:

PASS.

5.1.4.2 FMT_MTD.1/CoreData Management of TSF Data

5.1.4.2.1 FMT_MTD.1/CoreData TSS

Objective:

- The evaluator shall examine the TSS to determine that, for each administrative function identified in the AGD; 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.

- 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.

Evaluator Findings:

- The evaluator examined the TSS and determined that, **the only administrative function accessible prior to administrator log-in is the access banner displayed at each login prompt and no security functions are accessible.**
- If the TOE supports handling of X.509v3 certificates and implements a trust store, the evaluator examined the TSS and determined that it contains sufficient information to describe how the ability to manage the TOE's trust store is restricted. The TSS states:

  Only authorized administrators (those users assigned to the Sensor and Manager role "admin") can access this interface and use it to manage the configuration of the TOE (as enforced by the Identification and Authentication function), which includes management of the X.509v3 trust store.

Verdict:

PASS.

5.1.4.2.2 FMT_MTD.1/CoreData AGD

Objective:

- The evaluator shall review the AGD 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.
- If the TOE supports handling of X.509v3 certificates and provides a trust store, the evaluator shall review the AGD 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 AGD 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 AGD to determine that it explains how to designate a CA certificate a trust anchor.

Evaluator Findings:

- The evaluator reviewed the AGD section titled **'Handling user password between FIPS and non-FIPS images' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide and the sections 'Management of users and user roles', 'Managing devices', 'Monitoring Sensor Performance' and 'Custom Attack Definitions' of the IPS_11.1_Product_Guide** and determined 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.
- If the TOE supports handling of X.509v3 certificates and provides a trust store, the evaluator reviewed the AGD **'Managing certificates for manager and sensor' and the 'Syslog tab' of the IPS_11.1_Product_Guide** and determined 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 reviewed the AGD **'Managing certificates for manager and sensor' and the 'Syslog tab' of the IPS_11.1_Product_Guide** and determined that it provides sufficient information for the administrator to securely load CA certificates into the trust store.
- The evaluator also reviewed the AGD **'Managing certificates for manager and sensor' of the IPS_11.1_Product_Guide** and determined that it explains how to designate a CA certificate a trust anchor.

Verdict:

PASS.

5.1.4.3 FMT_SMF.1 Specification of Management Functions

5.1.4.3.1 FMT_SMF.1 TSS

Objective:

- The evaluator shall examine the TSS, the AGD 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).
- The evaluator shall examine the TSS and the AGD to verify they both describe the local administrative interface.
- The evaluator shall ensure the AGD includes appropriate warnings for the administrator to ensure the interface is local.
- 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 the AGD.
- The evaluator shall check that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and the AGD.

Evaluator Findings:

- The evaluator examined the TSS, the AGD and the TOE as observed during all other testing and confirmed that the management functions specified in FMT_SMF.1 are provided by the TOE.
- The evaluator confirmed that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface). The TSS states:

Accessibility of management functions via interfaces is detailed in the below table:

Management Functions	**Accessible interface (local, remote)**
Ability to administer the TOE locally and remotely	Both (CLI as well as GUI)
Ability to configure the access banner	Both (CLI as well as GUI)
Ability to configure the session inactivity time before session termination or locking	Both (CLI as well as GUI)
Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates	Both (CLI as well as GUI)
Ability to configure the authentication failure parameters for FIA_AFL.1	Both (CLI as well as GUI)
Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);	Both (CLI only) Note: This management function is only applicable to the Manager and not for the Sensors.
Ability to modify the behaviour of the transmission of audit data to an external IT entity	Remote (GUI only) Note: This management function is only applicable to the Manager and not for the Sensors.
Ability to configure the cryptographic functionality	Both (CLI as well as GUI)
Ability to configure the interaction between TOE components	Both (CLI as well as GUI)
Ability to re-enable an Administrator account	Remote (CLI only) Note: This management function is only applicable to the sensors and not for the Manager.
Ability to set the time which is used for timestamps	Remote (CLI only)
Ability to import X.509v3 certificates to the TOE's trust store	Remote (GUI only)
Ability to manage the trusted public keys database	Remote (CLI only)

- **All IPS-related management functions are accessible only via the remote GUI interface.** The evaluator examined the TSS and the AGD to verify they both describe the local administrative interface.
- The evaluator ensured the AGD includes appropriate warnings for the administrator to ensure the interface is local.
- The evaluator examined the TSS and ensured that the ways to configure the interaction between TOE components is detailed in the TSS and the AGD. TOE Summary Specification **FCO_CPC_EXT.1 and** the section **'Establishing Sensor-to-Manager communication' and 'Delete a Sensor from the Manager' under 'IPS Administration' of the IPS_11.1_Product_Guide and the 'Protocol features in the certified evaluated configuration' section of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** was used to determine the verdict of this assurance activity.
- The evaluator checked that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and the AGD.

Verdict:

PASS.

5.1.4.3.2 FMT_SMF.1/IPS Specification of Management Functions (IPS) TSS

Objective:

The evaluator shall verify that the TSS describes how the IPS data analysis and reactions can be configured. This may be performed in conjunction with the evaluation of IPS_ABD_EXT.1, IPS_IPB_EXT.1, and IPS_SBD_EXT.1.

Evaluator Findings:

The evaluator reviewed the TSS to ensure that it describes how the IPS data analysis and reactions can be configured **in conjunction with the evaluation of IPS_ABD_EXT.1, IPS_IPB_EXT.1, and IPS_SBD_EXT.1. TSS sections for these SFRs provide details of IPS data analysis and reactions.**

Verdict:

PASS.

5.1.4.3.3 FMT_SMF.1/IPS Specification of Management Functions (IPS) AGD

Objective:

The evaluator shall verify that the AGD describes the instructions for each function defined in the SFR, describes how to configure the IPS data analysis and reactions, including how to set any configurable defaults and how to configure each of the applicable analysis pattern matching methods and reaction modes.

Evaluator Findings:

The evaluator checked the AGD and ensured that it describes the instructions for each function defined in the SFR, describes how to configure the IPS data analysis and reactions, including how to set any configurable defaults and how to configure each of the applicable analysis pattern matching methods and reaction modes.

The section titled **'Manager Administration and IPS administration' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that **the AGD provides configurations on how to configure predefined filters and custom filters for IPS data analysis and how to configure the reaction modes.**

Verdict:

PASS.

5.1.4.4 FMT_SMR.2 Restrictions on Security Roles

5.1.4.4.1 FMT_SMR.2 TSS

Objective:

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.

Evaluator Findings:

The evaluator examined the TSS and determined that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE. The TSS states:

Only authorized administrators (those users assigned to the Sensor and Manager role "admin") can access this interface and use it to manage the configuration of the TOE (as enforced by the Identification and Authentication function), which includes management of the X.509v3 trust store.

Verdict:

PASS.

5.1.4.4.2 FMT_SMR.2 AGD

Objective:

The evaluator shall review the AGD 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.

Evaluator Findings:

The evaluator reviewed the AGD, examined the sections **'Protocol features in the certified evaluated configuration', 'Sensor CLI for Certification' and 'Manager configuration for Certification' in the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide and the section titled 'CLI commands' of the IPS_11.1_Product_Guide** and ensured 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.

Verdict:

PASS.

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

5.1.5.1.1 FPT_SKP_EXT.1 TSS

Objective:

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.

Evaluator Findings:

The evaluator examined the TSS, **FPT_APW_EXT.1/ FPT_SKP_EXT.1** and determined 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 describes how they are protected/obscured. The TSS states:

There is no filesystem access or administrative interface that allows any administrative users to read plaintext pre-shared keys, symmetric keys, and private keys. Error! Reference source not found. **and** Error! Reference source not found. **[ST] elaborate on how these keys are stored.**

Verdict:

PASS.

5.1.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

5.1.5.2.1 FPT_APW_EXT.1 TSS

Objective:

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.

Evaluator Findings:

The evaluator examined the TSS and determined 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 also details that 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. The TSS states:

There is no filesystem access or administrative interface that allows any administrative users to read plaintext secret and private keys.

The TOE stores administrative passwords in the database after adding a salt to the plaintext password and hashing the resultant value with HMAC-SHA-512.

Verdict:

PASS.

5.1.5.3 FPT_TST_EXT.1 TSF Testing

5.1.5.3.1 FPT_TST_EXT.1 TSS

Objective:

- 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.
- 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.

Evaluator Findings:

- The evaluator examined the TSS and ensured that it details the self-tests that are run by the TSF;.
- The evaluator ensured that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly. The TSS states:

  At power-on, a suite of known answer tests are performed on both, the Sensor and Manager devices, to confirm the correct operation of the cryptographic algorithms.

  As part of the known answer test, the TOE uses the algorithm to perform operation on a known value and compares it with a known result to verify the algorithm's correct operation. If any of the known answer test fails, then the TOE does not complete its bootup sequence and cryptographic functionalities will not be available.

Conditional self-tests are performed continuously during Manager and Sensor operations to confirm continued operation of the DRNG & NDRNG and sign/verify RSA and ECDSA pairwise consistency. In Continuous RBG test, every time a random number is generated by the DRBG, the TOE compares the current value with the previously generated value to ensure that the values are not same. If the values are same, then the value is discarded and a new random number is generated. If the DRBG continues to generate same value repeatedly then the DRBG is considered "stuck" and the TOE enter an error state where the cryptographic functionalities will not be available.

As part of the pairwise consistency test, every time the TOE generates a public-private keypair, the TOE perform encrypt and decrypt operations on a sample data to verify that the key-pair was generated correctly.

Entropy health testing is performed at start-up and continuously during operation. Then a continuous RBG test is performed each time random data is requested. If the test fails due to insufficient entropy in the pool then the function does not provide the entropy, backs off, and retries again giving time for additional entropy to be collected.

On initiation of a trusted update, both Sensor and Manager devices verify the integrity of all firmware modules using RSA 2048 bit key with SHA-256 signature. This includes testing of every component in the image. The kernel and the rest of the components are verified with an RSA 2048 bit with SHA-256 signature. If the integrity test fails, then the TOE does not complete its bootup sequence and cryptographic functionalities will not be available.

- The evaluator examined the TSS and ensured that it details which TOE component performs which self-tests and when these self- tests are run.

The evaluator found that **both Sensor and Manager devices perform cryptographic self-tests and entropy health-tests during power-on, with the latter also being performed periodically during normal operation. These ensure accurate operation of the cryptographic functionalities. Both devices also perform module-wise software integrity tests during initiation of a trusted update, thereby confirming the modules have not been modified or replaced in any unauthorized manner.**

Verdict:

PASS.

5.1.5.3.2 FPT_TST_EXT.1 AGD

Objective:

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

Evaluator Findings:

- The evaluator also ensured that the AGD describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors correspond to those described in the TSS.
- The evaluator examined the AGD sectioned titled **'Device bootup processing' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** and ensured that it describes how to determine from an error message returned which TOE component has failed the self-test.

Verdict:

PASS.

5.1.5.4 FPT_TUD_EXT.1 Trusted Update

5.1.5.4.1 FPT_TUD_EXT.1 TSS

Objective:

- The evaluator shall verify that the TSS describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the TSS needs to describe how and when the inactive version becomes active.
- The evaluator shall verify this description.
- 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.
- 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.
- 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 AGD. In that case the evaluator should examine the AGD instead.
- 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.

Evaluator Findings:

- The evaluator verified that the TSS describes how to query the currently active version.
- The evaluator verified this description. The TSS states:

  Following successful authentication authorized administrators can perform management actions such as query the current version of the TOE software on Manager and the Sensor(s) using CLI commands or version information visible via the GUI.
- The evaluator verified 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 verified that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails.

- The evaluator verified that the TSS describes the method by which the digital signature is verified to included and how the candidate updates are obtained, the processing associated with verifying the digital signature of the update, and the actions that take place for both successful and unsuccessful signature verification,

  As per ST, 'The images are signed with a Trellix key. Once the image has been downloaded, the TOE checks the signature of the image (against the Trellix public key stored in a file in the internal media) before the image is applied, and does not proceed with the installation in case of a failure'.
- The options 'support automatic checking for updates' or 'support automatic updates' are not chosen from the selection in FPT_TUD_EXT.1.2, so not applicable.
- The evaluator examined the TSS and ensured 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 is performed for each TOE component. Alternatively, this description can be provided in the AGD. In that case the evaluator should examine the AGD instead.

The evaluator found that the TSS states that: **Following successful authentication authorized administrators can perform management actions such as query the current version of the TOE software on Manager and the Sensor(s) using CLI commands or version information visible via the GUI. The administrator can initiate an update of the TOE software using an image file hosted on a SCP Server, via the Manager Web UI or the Manager/sensor CLI. Sensor updates initiated through the Manager Web UI or information regarding Manager image updates are pushed out to connected Sensors of the intra-TSF trusted channel and vice versa. System functions, including the sensor-Manager channel cease functioning for the duration following the update, and during the reboot that follows. This channel can be re-established by following the steps in the guidance document, while other functionalities come back up automatically, shortly after the reboot. The images are signed with a Trellix key. Once the image has been downloaded, the TOE checks the signature of the image (against the Trellix public key stored in a file in the internal media) before the image is applied, and does not proceed with the installation in case of a failure.**

- TOE does not claim 'Published hash'.

Verdict:

PASS.

5.1.5.4.2 FPT_TUD_EXT.1 AGD

Objective:

- The evaluator shall verify that the AGD describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the AGD needs to describe how to query the loaded but inactive version.
- The evaluator shall verify that the AGD 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.
- If a published hash is used to protect the trusted update mechanism, the evaluator shall verify that the AGD describes how the Security Administrator can obtain authentic published hash values for the updates.
- For distributed TOEs the evaluator shall verify that the AGD 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 AGD 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.

- If this was information was not provided in the TSS: For distributed TOEs, the evaluator shall examine the AGD 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.

Evaluator Findings:

- The evaluator verified that the AGD section titled ''describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the AGD describes how to query the loaded but inactive version.
- The evaluator verified that the AGD describes how the verification of the authenticity of the update is performed (digital signature verification or verification of published hash). The description includes the procedures for successful and unsuccessful verification. The description corresponds to the description in the TSS.
- TOE does not claim 'Published hash'.
- The evaluator examined the AGD and ensured that it 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.
- This information was provided in the TSS: For distributed TOEs, the evaluator examined the AGD and ensured 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 relevant information is found in the following section(s):

The **command 'show ManagerVersion' under section 'Manager Shell Commands' and command 'show' under section 'IPS CLI Commands - Normal Mode'** of the **IPS_11.1_Product_Guide** describes how to query the currently active version for the Manager and Sensor respectively.

The section **'Verification of authenticity of the software images'** of the **Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** describes how the verification of the authenticity of the update is performed.

The sections titled **'Device software' under 'Trellix IPS Protection Status', 'Upgrade' under 'Manager Shell Commands', 'loadimage' and 'loadsavedimage' under 'IPS Commands - Normal Mode'** of the **IPS_11.1_Product_Guide** describe downloading/deploying of sensor software via GUI, the process for upgrading the manager software via CLI, and the process for upgrading the sensor software using the sensor CLI respectively**.** These sections as mentioned were used to determine the verdict of this assurance activity.

Verdict:

PASS.

5.1.5.5 FPT_STM_EXT.1 Reliable Time Stamps

5.1.5.5.1 FPT_STM_EXT.1 TSS

Objective:

- 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.
- 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.

Evaluator Findings:

- The evaluator examined the TSS and ensured 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.

The relevant information is found in the following section(s): TOE Summary Specification **FPT_STM_EXT.1**.

Upon investigation, the evaluator found that the TSS states that**: The NS sensor platforms maintain a system clock used to provide date/time details for use by the TOE. The Manager periodically passes a timestamp reference to the Sensors to ensure clocks within an IPS system are consistent. This occurs on power up when establishing the TLS crypto channels to Manager, upon every TLS reestablishment due to link/network issues to Manager and when a TLS reconnection is initiated by the administrator. This timestamp is sent by the Manager over a TLS connection. The administrator manually sets the time on the Manager to keep the time in sync with the outside world.**

Each Sensor uses this timestamp to synchronize its own independent timing mechanism synchronizing at regular intervals per the timestamps sent from the Manager management platform.

The system clock is used by the Sensor and Manager to timestamp all audit events recorded in the audit log, as identified in Security Audit(FAU_GEN.1) section of TSS. Additionally, it is also used as a source of clock cycles which are used to implement timers for functionalities such as inactivity and authentication failure timeouts, rekeying etc. Other functionalities making use of time such as X.509v3 certificate validation (eg: for certificate expiry/revocation checks), TLS and SSH session times etc. also make use of the system clock.

- ST does not select claim “obtain time from the underlying virtualization system”.

Verdict:

PASS.

5.1.5.5.2 FPT_STM_EXT.1 AGD

Objective:

- The evaluator examines the AGD to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the AGD 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.
- If the TOE supports obtaining time from the underlying VS, the evaluator shall verify the AGD specifies any configuration steps necessary. If no configuration is necessary, no statement is necessary in the AGD. 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 AGD informs the administrator of the maximum possible delay.

Evaluator Findings:

- The evaluator examined the AGD and ensured that it instructs the administrator how to set the time.

The relevant information is found in the following section(s): The **command ‘set time’ under the section ‘Manager shell commands’ of the IPS_11.1_Product_Guide and the section ‘Protocol features in the certified evaluated configuration’ of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found through the section **‘Protocol features in the certified evaluated configuration’ of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide states that: Usage of NTP is not permitted. The system time may be configured by authorized administrators via the “timedatectl” command of the Manager CLI.**

The command ‘set time’ under the section ‘Manager shell commands commands’ of the IPS_11.1_Product_Guide provides instructions for configuring time on the TOE.

- TOE does not support obtaining time from the underlying VS.

Verdict:

PASS.

5.1.6 TOE Access (FTA)

5.1.6.1 FTA_SSL_EXT.1 TSF-Initiated Session Locking

5.1.6.1.1 FTA_SSL_EXT.1 TSS

Objective:

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.

Evaluator Findings:

The evaluator examined the TSS and determined that it details whether local administrative session locking or termination is supported and the related inactivity time period settings.

The relevant information is found in the following section(s): TOE Summary Specification **FTA_SSL_EXT.1**.

Upon investigation, the evaluator found that the TSS states that: **Following an administrator configured period of inactivity (of both local and remote sessions) the session will be terminated, requiring re-authentication by the administrator before the access to TOE functionality can be gained.**

Verdict:

PASS.

5.1.6.1.2 FTA_SSL_EXT.1 AGD

Objective:

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

Evaluator Findings:

The evaluator confirmed that the AGD states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period.

The relevant information is found in the following section(s):

The sections **'Configure session control settings', command 'set console timeout'** under **'IPS CLI Commands - Normal Mode'** and the section **'Customize inactivity timeout via the Manager shell'** under **'Configure session control settings'** of the **IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the **AGD provides instructions for configuring the inactivity time period and hence is able to set the time for the local session expiry.**

Verdict:

PASS.

5.1.6.2 FTA_SSL.3 TSF-Initiated Termination

5.1.6.2.1 FTA_SSL.3 TSS

Objective:

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

Evaluator Findings:

The evaluator examined the TSS and determined that it details the administrative remote session termination and the related inactivity time period.

The relevant information is found in the following section(s): TOE Summary Specification **FTA_SSL.3.**

Upon investigation, the evaluator found that the TSS states that: **Following an administrator configured period of inactivity (of both local and remote sessions) the session will be terminated, requiring re-authentication by the administrator before the access to TOE functionality can be gained.**

Verdict:

PASS.

5.1.6.2.2 FTA_SSL.3 AGD

Objective:

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

Evaluator Findings:

The evaluator confirmed that the AGD includes instructions for configuring the inactivity time period for remote administrative session termination.

The relevant information is found in the following section(s):

The **section 'Configure session control settings' and command 'set console timeout' under 'IPS CLI Commands - Normal Mode' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the **AGD provides instructions for configuring the inactivity time period and hence is able to set the time for the remote session expiry.**

Verdict:

PASS.

5.1.6.3 FTA_SSL.4 User-Initiated Termination

5.1.6.3.1 FTA_SSL.4 TSS

Objective:

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

Evaluator Findings:

The evaluator examined the TSS and determined that it details how the local and remote administrative sessions are terminated.

The relevant information is found in the following section(s): TOE Summary Specification **FTA_SSL.4.**

Upon investigation, the evaluator found that the TSS states that: **The administrator can issue an "exit" command for the CLI or click on the logout button within the GUI to terminate their session once they have completed all administrative tasks.**

Verdict:

PASS.

5.1.6.3.2 FTA_SSL.4 AGD

Objective:

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

Evaluator Findings:

The evaluator confirmed that the AGD states how to terminate a local or remote interactive session.

The relevant information is found in the following section(s):

The **commands 'exit' under 'IPS CLI Commands - Normal Mode', 'exit' under 'Manager Shell Commands' and section 'Menu bar' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the **AGD provides instructions on how to terminate a local or remote interactive session.**

Verdict:

PASS.

5.1.6.4 FTA_TAB.1 Default TOE Access Banners

5.1.6.4.1 FTA_TAB.1 TSS

Objective:

- 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).

Evaluator Findings:

- The evaluator checked the TSS and ensured that it details each administrative method of access (local and remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS).
- The evaluator checked the TSS and ensured 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).

The relevant information is found in the following section(s): TOE Summary Specification **FTA_TAB.1.**

Upon investigation, the evaluator found that the TSS states that: **All management of the TOE is performed though the Web UI of Manager component, and CLI of individual components (Manager and Sensor). Identification and authentication are required for both local and remote administrator access. Remote access to the TOE is via an SSH or HTTPS session (from the Management Workstation) and local access to the TOE is via the appliance console port.**

Prior to logon via console, SSH, and web GUI, a consent banner is displayed to the administrator warning that proceeding with authentication is consenting to the terms of use of the TOE. The TOE banner for both Sensor and Manager devices is administrator configurable via the Manager. Having acknowledged the access banner, the user is then prompted to enter their username and password. The TOE supports local authentication where it looks up the username in /etc/passwd and compares the hash of the password to the value in /etc/shadow. If the credentials correspond to an entry in the files, the user is successfully authenticated and is authorized to access the management interface.

Verdict:

PASS.

5.1.6.4.2 FTA_TAB.1 AGD

Objective:

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

Evaluator Findings:

The evaluator examined the AGD and ensured that it details each administrative method of access (local and remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS), lists all administrative methods of access available to the Security Administrator, and that states that the TOE displays an advisory notice and a consent warning message for each administrative method of access.

The relevant information is found in the following section(s):

The command **'Set login banner' under 'CLI commands', sections, 'Add a Manager logon banner' for the manager and section 'Add a device logon banner' for the sensor of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found **the guidance describes setting the login banner (text, image, etc.) on both the GUI and CLI.**

Verdict:

PASS.

5.1.7 Trusted Path (FTP)

5.1.7.1 FTP_ITC.1 Inter-TSF Trusted Channel

5.1.7.1.1 FTP_ITC.1 TSS

Objective:

- 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.

Evaluator Findings:

- The evaluator examined the TSS and determined 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 relevant information is found in the following section(s): TOE Summary Specification **FTP_ITC.1.**

  Upon investigation, the evaluator found that the TSS states that: **The TOE operates as an SSH Server and Client. The SSH Client functionality is used by a Sensor to provide a trusted channel with the SCP server for trusted updates. The SSH Server functionality is used to provide a trusted path for administrative access for Manager as well as Sensors. The TOE (Manager) operates as TLS Client to provide a trusted channel with the syslog server.**

- Next, the evaluator verified that for each communication identified in the TSS a description of the secure communication mechanisms is provided. Specifically, the evaluator found that the following protocols are used to connect to authorized IT entities: **TLS, SSH.**

  The evaluator also confirmed 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 relevant information is found in the following section(s): TOE Summary Specification **FTP_ITC.1.**

  Upon investigation, the evaluator found that the TSS states that: **The TOE (both Manager and Sensor) operates as an SSH Server with the following algorithm support:**

  - **Version: v2**
  - **Cipher/MAC: aes128-gcm@openssh.com, aes256-gcm@openssh.com**
  - **Hostkey: ecdsa-sha2-nistp256**
  - **Key Exchange: ecdh-sha2-nistp256**
  - **Authentication: Password, ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256**

  The TOE (Sensor) operates as an SSH Client with the following algorithm support:

  - **Version: v2**
  - **Cipher/MAC: aes128-gcm@openssh.com, aes256-gcm@openssh.com**
  - **Hostkey: ecdsa-sha2-nistp256, ssh-rsa, rsa-sha2-256, rsa-sha2-512**
  - **Key Exchange: ecdh-sha2-nistp256**
  - **Authentication: Password, ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256**

  The TOE (Manager) operates as TLS Client to provide a trusted channel with the syslog server. This client supports TLSv1.2 with the following ciphersuites:

  - **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256**
  - **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384**

  Key Establishment is performed using Elliptic Curve Diffie-Hellman P-256 keys.

  Next, the evaluator compared the list identified in the TSS to the definition of the SFR in ST. The evaluator found the identified protocols to be consistent.

Verdict:

PASS.

5.1.7.1.2 FTP_ITC.1 AGD

Objective:

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

Evaluator Findings:

The evaluator confirmed that the AGD contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

The relevant information is found in the following section(s):

The sections **'Sensor features in FIPS compliant images' and 'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide and the command 'ssh', 'upgrade' and 'loadimage' under 'CLI commands', and section 'Alert notification options' of the IPS_11.1_Product_Guide** were used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD activity states that: **AGD provides configuration instruction for configuring connections with each authorized IT entity. Specifically, the evaluator found that AGD provides guidance for configuring connections with the following authorized IT entities: audit server.**

Next, the evaluator reviewed AGD and found that **for each connection a description of how to recover from unintentional disconnections. Once the disruption has been corrected, the syslog client on the TOE will automatically attempt to re-negotiate the TLS channel upon the next retry.**

Verdict:

PASS.

5.1.7.2 FTP_TRP.1/Admin Trusted Path

5.1.7.2.1 FTP_TRP.1/Admin TSS

Objective:

- 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.

Evaluator Findings:

- The evaluator examined the TSS and determined that the methods of remote TOE administration are indicated, along with how those communications are protected.

  The relevant information is found in the following section(s): TOE Summary Specification **FTP_TRP.1/Admin.**

  Upon investigation, the evaluator found that **the TSS identifies each method of remote TOE administration along with how the communications are protected. Specifically, the evaluator found that the TSS identifies the following methods of remote administration a corresponding protection,**
    - **Remote CLI over SSH**
    - **Remote GUI over HTTPS/TLS**
- The evaluator also confirmed 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.

  The relevant information is found in the following section(s): TOE Summary Specification **FTP_TRP.1/Admin.**

  Upon investigation, the evaluator found that the **TSS identifies the following protocols used for remote administration,**
    - **HTTPS/TLS**
    - **SSH**

  Next, the evaluator compared the protocols identified in the TLS to the definition of the SFR. The evaluator found that the protocols listed in the TSS are consistent with the protocols listed in the definition of the SFR.

Verdict:

PASS.

5.1.7.2.2 FTP_TRP.1/Admin AGD

Objective:

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

Evaluator Findings:

The evaluator confirmed that the AGD contains instructions for establishing remote administrative sessions for each supported method.

The relevant information is found in the following section(s):

The section **'Accessing the Manager from a client machine' and 'ssh' under 'CLI commands' of the IPS_11.1_Product_Guide** were used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD provides **instructions for the configuration of the protocols used to secure remote administrative session. Specifically, AGD provides instructions for configuring the following protocols:**

- **HTTPS/TLS**
- **SSH**

Verdict:

PASS.

5.1.8 Intrusion Prevention System (IPS)

5.1.8.1 Anomaly-Based IPS Functionality (IPS_ABD_EXT)

5.1.8.1.1 IPS_ABD_EXT.1 Anomaly-Based IPS Functionality TSS

Objective:

- The evaluator shall verify that the TSS describes the composition, construction, and application of baselines or anomaly-based attributes specified in IPS_ABD_EXT.1.1.
- The evaluator shall verify that the TSS provides a description of how baselines are defined and implemented by the TOE, or a description of how anomaly-based rules are defined and configured by the administrator.
- If 'frequency' is selected in IPS_ABD_EXT.1.1, the TSS shall include an explanation of how frequencies can be defined on the TOE.
- If 'thresholds' is selected in IPS_ABD_EXT.1.1, the TSS shall include an explanation of how the thresholds can be defined on the TOE.
- The evaluator shall verify that each baseline or anomaly-based rule can be associated with a reaction specified in IPS_ABD_EXT.1.3.
- The evaluator shall verify that the TSS identifies all interface types capable of applying baseline or anomaly-based rules and explains how they are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that it describes the composition, construction, and application of baselines or anomaly-based attributes specified in IPS_ABD_EXT.1.1.

- The evaluator reviewed the TSS to ensure that it provides a description of how baselines are defined and implemented by the TOE, or a description of how anomaly-based rules are defined and configured by the administrator.
- Since 'frequency' is selected in IPS_ABD_EXT.1.1, the evaluator reviewed the TSS to ensure that, the TSS includes an explanation of how frequencies can be defined on the TOE.
- The evaluator verified that 'thresholds' is not selected in IPS_ABD_EXT.1.1 SFR.
- The evaluator reviewed the TSS to ensure that each baseline or anomaly-based rule can be associated with a reaction specified in IPS_ABD_EXT.1.3.
- The evaluator reviewed the TSS to ensure that it identifies all interface types capable of applying baseline or anomaly-based rules and explains how they are associated with distinct network interfaces.

The relevant information is found in the following section(s): TOE Summary Specification **IPS_ABD_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TSF implements anomaly detection by comparing protocol headers against the underlying RFC specifications. The per-protocol specification provides pre-packaged signatures and rules to support the same. A user can load the protocol specification with additional signatures, to detect for specific anomalies and deployment specific requirements, using the UDS (User Defined Signatures) framework. The UDS tool allows a user to add, delete and modify signatures to tailor their Sensors for best detection efficacy matching their network traffic and provide the desired response actions. The above policies can be selectively applied to different interfaces, port clusters etc. as explained in the guidance document. Frequency in terms of number of occurences per 'n' seconds can be configured for attack definitions under an IPS policy. Under the attack definition in an IPS policy, the TSF can be configured to generate an alert and allow the traffic flow, send a TCP reset to the source, send a TCP reset to the destination, send an ICMP host unreachable, or block the traffic when an anomaly is detected.**

Verdict:

PASS.

5.1.8.1.2 IPS_ABD_EXT.1 Anomaly-Based IPS Functionality AGD

Objective:

- The evaluator shall verify that the AGD provides instructions to manually create baselines or anomaly-based rules according to the selections made in IPS_ABD_EXT.1.1. Note that dynamic "profiling" of a network to establish a baseline is outside the scope of the PP-Module.
- The evaluator shall verify that the AGD provides instructions to associate reactions specified in IPS_ABD_EXT.1.3 with baselines or anomaly-based rules.
- The evaluator shall verify that the AGD provides instructions to associate the different policies with distinct network interfaces.

Evaluator Findings:

- The evaluator checked the AGD and ensured that it provides instructions to manually create baselines or anomaly-based rules according to the selections made in IPS_ABD_EXT.1.1.
- The evaluator checked the AGD and ensured that it provides instructions to associate reactions specified in IPS_ABD_EXT.1.3 with baselines or anomaly-based rules.
- The evaluator checked the AGD and ensured that it provides instructions to associate the different policies with distinct network interfaces.

The section **'Preconfigured policies', 'Configure attack details', 'View attack description' and 'Response management' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD **describes how to create anomaly-based rules and the reactions to those rules, along with instructions to associate the different policies with distinct network interfaces.**

The 'Default Prevention' policy category listed in the AGD contains pre-configured IPS policies that allow an administrator to deploy baselines or anomaly-based rules.

The 'Settings' Tab can be used to configure attack related parameters such as frequency, response actions and logging related actions. The 'Description' tab provides additions details about the attack such as category, the signature used, etc.

The claimed response actions such as blocking/dropping, sending a TCP reset or ICMP unreachable messages, etc. are also described in the AGD.

Verdict:

PASS.

5.1.8.2 IP Blocking (IPS_IPB_EXT)

5.1.8.2.1 IPS_IPB_EXT.1 IP Blocking TSS

Objective:

- The evaluator shall verify how good/bad lists affect the way in which traffic is analyzed with respect to processing packets.
- The evaluator shall also verify that the TSS provides details for the attributes that create a known good list, a known bad list, and their associated rules, including how to define the source or destination IP address (e.g. a single IP address or a range of IP addresses).
- If the TSF uses address types other than a single IP or a range of IP addresses (e.g. MAC addresses), the evaluator shall check that the TSS explains what configurations would cause non-IP lists of known-good and known-bad addresses to take precedence over IP-based address lists.
- The evaluator shall also verify that the TSS identifies all the roles and level of access for each of those roles that have been specified in the requirement.

Evaluator Findings:

- The evaluator reviewed the TSS to verify how good/bad lists affect the way in which traffic is analyzed with respect to processing packets.
- The evaluator reviewed the TSS to ensure that it provides details for the attributes that create a known good list, a known bad list, and their associated rules, including how to define the source or destination IP address (e.g. a single IP address or a range of IP addresses).
- TSF does not uses any additional address types.
- The evaluator reviewed the TSS to ensure that it identifies all the roles and level of access for each of those roles that have been specified in the requirement.

The relevant information is found in the following section(s): TOE Summary Specification **IPS_IPB_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The IPS Administrator ('admin' user) can configure the Sensor for good/bad IP lists via the Firewall Policy Configuration page on the Manager, which allows them to specify the firewall rules. Each rule can include good/bad IP address and/or range definitions with an associated response action. Good lists can be defined by specifying the IP address/range and setting the response action as 'Scan'. Similarly bad lists will have the response action set to 'Drop'. The TOE compares source/destination IPs of all traffic against active Firewall policies and processes them based on the set response action. Non-IP lists are not supported by the TOE.**

Verdict:

PASS.

5.1.8.2.2 IPS_IPB_EXT.1 IP Blocking AGD

Objective:

- The evaluator shall verify that the AGD provides instructions with how each role specified in the requirement can create, modify and delete the attributes of a known good and known bad lists.
- If the TSF uses address types other than a single IP or a range of IP addresses (e.g. MAC addresses), the evaluator shall check that the AGD includes instructions for any configurations that would cause non-IP lists of known-good and known-bad addresses to take precedence over IP-based address lists.

Evaluator Findings:

- The evaluator checked the AGD and ensured that it provides instructions with how each role specified in the requirement can create, modify and delete the attributes of a known good and known bad lists.
- The evaluator checked the AGD and ensured that, if the TSF uses address types other than a single IP or a range of IP addresses (e.g. MAC addresses), the AGD includes instructions for any configurations that would cause non-IP lists of known-good and known-bad addresses to take precedence over IP-based address lists.

The section **'Create Firewall policies' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD **provides instructions on how to configure a firewall policy using access rules as the building blocks. The 'Policy' tab can be used to configure the firewall policies. After creating the firewall policy, it can be assigned to the required Sensor resource. The section also provides a step-by-step guide on how to configure the firewall feature and set a response action for the specified access rule. Actions such as deny, drop, ignore, scan, etc. can be configured.**

Verdict:

PASS.

5.1.8.3 Network Traffic Analysis (IPS_NTA_EXT)

5.1.8.3.1 IPS_NTA_EXT.1 Network Traffic Analysis IPS_NTA_EXT.1.1 TSS

Objective:

- The evaluator shall verify that the TSS explains the TOE's capability of analyzing IP traffic in terms of the TOE's policy hierarchy (precedence). The TSS should identify if the TOE's policy hierarchy order is configurable by the administrator for IPS policy elements (known-good lists, known-bad lists, signature-based rules, and anomaly-based rules).
- Regardless of whether the precedence is configurable, the evaluator shall verify that the TSS describes the default precedence as well as the IP analyzing functions supported by the TOE.

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that it explains the TOE's capability of analyzing IP traffic in terms of the TOE's policy hierarchy (precedence) and that it identifies if the TOE's policy hierarchy order is configurable by the administrator for IPS policy elements (known-good lists, known-bad lists, signature-based rules, and anomaly-based rules).
- The evaluator reviewed the TSS to ensure that it describes the default precedence as well as the IP analyzing functions supported by the TOE.

The relevant information is found in the following section(s): TOE Summary Specification **IPS_NTA_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **By default, the TSF applies rules for all IPS policy elements in the order they are configured. Precedence for rules within firewall policies used for IPS_IPB_EXT.1 can be manually re-adjusted by the administrator even after definition.**

Verdict:

PASS.

5.1.8.3.2 IPS_NTA_EXT.1 Network Traffic Analysis IPS_NTA_EXT.1.1 AGD

Objective:

- The evaluator shall verify that the AGD describes the default precedence.
- If the precedence is configurable, the evaluator shall verify that the AGD explains how to configure the precedence.

Evaluator Findings:

- The evaluator checked the AGD and ensured that it describes the default precedence.
- The evaluator checked the AGD and ensured that, if the precedence is configurable, the AGD explains how to configure the precedence.

The section **'Create Firewall policies' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD states that: **the Sensor executes the rules in a top-down fashion and stops the execution when a rule matches.**
The AGD also provides instructions to insert a new rule above or below an existing rule and also to move a rule upwards or downwards in the list using appropriate buttons in the 'Access Rules' tab.

Verdict:

PASS.

5.1.8.3.3 IPS_NTA_EXT.1.2 TSS

Objective:

- The evaluator shall verify that the TSS indicates that the following protocols are supported:
  - IPv4
  - IPv6
  - ICMPv4
  - ICMPv6
  - TCP
  - UDP

The evaluator shall verify that the TSS describes how conformance with the identified protocols has been determined by the TOE developer (e.g., third party interoperability testing, protocol compliance testing).

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that it indicates that the following protocols are supported:

  IPv4
  IPv6
  ICMPv4
  ICMPv6
  TCP
  UDP

The evaluator reviewed the TSS to ensure that it describes how conformance with the identified protocols has been determined by the TOE developer (e.g., third party interoperability testing, protocol compliance testing).

The relevant information is found in the following section(s): TOE Summary Specification **IPS_NTA_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TSF allows Sensor traffic interfaces to be configured into promiscuous mode, inline mode, or response mode to support network traffic analysis. The TSF supports the following protocols for analysis:**

- **IPv4**
- **IPv6**
- **ICMPv4**
- **ICMPv6**
- **TCP**
- **UDP**

Conformance to the RFCs corresponding to the above protocols is demonstrated by protocol compliance testing by the product QA team.

Verdict:

PASS.

5.1.8.3.4 IPS_NTA_EXT.1.3 TSS

Objective:

- The evaluator shall verify that the TSS identifies all interface types capable of being deployed in the modes of promiscuous, and or inline mode as well as the interfaces necessary to facilitate each deployment mode (at a minimum, the interfaces need to support inline mode).
- The evaluator shall also check that the TSS provides a description for how the management interface is logically distinct from any sensor interfaces.

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that it identifies all interface types capable of being deployed in the modes of promiscuous, and or inline mode as well as the interfaces necessary to facilitate each deployment mode (at a minimum, the interfaces need to support inline mode).
- The evaluator also checked that the TSS provides a description for how the management interface is logically distinct from any sensor interfaces.

The relevant information is found in the following section(s): TOE Summary Specification **IPS_NTA_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TSF allows Sensor traffic interfaces to be configured into promiscuous mode, inline mode, or response mode to support network traffic analysis.**
The sensor has a distinct/dedicated management interface which is physically, and as a consequence logically distinct from other sensor interfaces. IPS functionalities work in conjunction with dedicated 'monitoring ports' and have no association with the management port/interface.

Verdict:

PASS.

5.1.8.3.5 IPS_NTA_EXT.1.3 AGD

Objective:

- The evaluator shall verify that the AGD provides instructions on how to deploy each of the deployment methods outlined in the TSS.

- The evaluator shall also verify that the AGD provides instructions of applying IPS policies to interfaces for each deployment mode.
- If the management interface is configurable, the evaluator shall verify that the AGD explains how to configure the interface as a management interface.
- The evaluator shall verify that the AGD explains how the TOE sends commands to remote traffic filtering devices if this functionality is supported.

Evaluator Findings:

- The evaluator checked the AGD and ensured that it provides instructions on how to deploy each of the deployment methods outlined in the TSS.
- The evaluator checked the AGD and ensured that it provides instructions of applying IPS policies to interfaces for each deployment mode.
- The evaluator checked the AGD and ensured that it, if the management interface is configurable, the AGD explains how to configure the interface as a management interface.
- The evaluator checked the AGD and ensured that it explains how the TOE sends commands to remote traffic filtering devices if this functionality is supported.

The AGD sections **'Monitoring port details', 'Change a monitoring port from SPAN mode to TAP or Inline mode (and vice versa)', 'View management port settings', 'Configuration of device monitoring and response ports' and 'Assign IPS policy to interfaces and subinterfaces' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the **AGD provides specific instructions to deploy the dedicated IPS monitoring ports in the selected deployment methods, with 'SPAN' mode being promiscuous mode (listen-only) and 'Inline Fail Open' and 'Inline Fail Closed' modes being the Inline mode (data pass-through). Response ports fall under the session-reset capable category along with the inline mode ones.**
In addition, the evaluator found that NS Sensors have a management interface with details visible in a separate 'Management Port' tab. The management interface is dedicated and not configurable.
The TOE does not support sending commands to remote traffic filtering devices.

Verdict:

PASS.

5.1.8.4 Signature-Based IPS Functionality (IPS_SBD_EXT)

5.1.8.4.1 IPS_SBD_EXT.1 Signature-Based IPS Functionality

5.1.8.4.2 IPS_SBD_EXT.1.1 TSS

Objective:

- The evaluator shall verify that the TSS describes what is comprised within a signature rule.
- The evaluator shall verify that each signature can be associated with a reaction specified in IPS_SBD_EXT.1.5.
- The evaluator shall verify that the TSS identifies all interface types that are capable of applying signatures and explains how rules are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that it describes what is comprised within a signature rule.

- The evaluator reviewed the TSS to ensure that each signature can be associated with a reaction specified in IPS_SBD_EXT.1.5.
- The evaluator reviewed the TSS to ensure that it identifies all interface types that are capable of applying signatures and explains how rules are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

The relevant information is found in the following section(s): TOE Summary Specification **IPS_SBD_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TSF supports signature-based traffic analysis. A signature rule comprises of multiple components. It can specify a pattern that denotes a literal string or a regular expression. They are typically represented as protocol-specific-string fields and/or numeric-fields that encompass numerical matches, like return codes and TLV encodings. The physical and logical scope to which these attacks and policies are applied is also specified in the signature file. This can be one/more physical interfaces, sub-interfaces, CIDR-blocks, VLANs and port cluster constructs. Specifically, each attack is associated with its signature and a response action (from the ones mentioned under IPS_SBD_EXT.1.5). Together, they define a policy that can be applied to the monitoring ports/interfaces, as desired by the user.**

Verdict:

PASS.

*5.1.8.4.3 IPS_SBD_EXT.1.1 AGD **[TD0722 applied]***

Objective:

- The evaluator shall verify that the operational guidance provides instructions with how to create and/or configure rules using the following protocols and header inspection fields:
  - IPv4: version; header length; packet length; ID; IP flags; fragment offset; time to live (TTL); protocol; header checksum; source address; destination address; IP options; and, if selected, type of service (ToS).
  - IPv6: Version; payload length; next header; hop limit; source address; destination address; routing header; and, if selected, traffic class and/or flow label.
  - ICMP: type; code; header checksum; and, if selected, other header fields (varies based on the ICMP type and code).
  - ICMPv6: type; code; and header checksum.
  - TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.
  - UDP: source port; destination port; length; and UDP checksum.

Evaluator Findings:

- The evaluator checked the AGD and ensured that it provides instructions with how to create and/or configure rules using the following protocols and header inspection fields:
  - IPv4: version; header length; packet length; ID; IP flags; fragment offset; time to live (TTL); protocol; header checksum; source address; destination address; and IP options.
  - IPv6: Version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options.
  - ICMP: type; code; header checksum; and rest of header (varies based on the ICMP type and code).
  - ICMPv6: type; code; and header checksum.
  - TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.
  - UDP: source port; destination port; length; and UDP checksum.

The sections **'Create custom attacks', 'Importing Trellix IPS-defined custom attacks', 'Create an exploit attack without template', 'Create a signature' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD **provides instructions for configuring and importing rules using protocols and header inspection fields. An end user can also import vendor-provided custom attacks as per their protocols and header inspection requirements. The 'Custom Exploit (Signature-Based)' detection type can be used to define custom attacks for the mentioned fields with the 'String Pattern Match', 'Numeric Value Match' and 'Single Fixed Field Match' comparison types. The 'Create the signature for single fixed field match' section provides an example for defining a destination IP (IPv4) based custom attack.**

Verdict:

PASS.

5.1.8.4.4 IPS_SBD_EXT.1.2 TSS

Objective:

- The evaluator shall verify that the TSS describes what is comprised within a string-based detection signature.
- The evaluator shall verify that each packet payload string-based detection signature can be associated with a reaction specified in IPS_SBD_EXT.1.5.

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that it describes what is comprised within a string-based detection signature.
- The evaluator reviewed the TSS to ensure that each packet payload string-based detection signature can be associated with a reaction specified in IPS_SBD_EXT.1.5.

The relevant information is found in the following section(s): TOE Summary Specification **IPS_SBD_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TSF supports signature-based traffic analysis. A signature rule comprises of multiple components. It can specify a pattern that denotes a literal string or a regular expression. They are typically represented as protocol-specific-string fields and/or numeric-fields that encompass numerical matches, like return codes and TLV encodings. The signature file pushed to the Sensor from the Manager contains attack definitions and policies for their detection. The physical and logical scope to which these attacks and policies are applied is also specified in the signature file.**

Verdict:

PASS.

5.1.8.4.5 IPS_SBD_EXT.1.2 AGD

Objective:

- The evaluator shall verify that the AGD provides instructions with how to configure rules using the packet payload string-based detection fields defined in IPS_SBD_EXT.1.2.
- The evaluator shall verify that the AGD provides instructions with how to configure reactions specified in IPS_SBD_EXT.1.5 for each string-based detection signature.
- The evaluator shall verify that the AGD provides instructions with how rules are associated with distinct network interfaces that are capable of being associated with signatures.

Evaluator Findings:

- The evaluator checked the AGD and ensured that it provides instructions with how to configure rules using the packet payload string-based detection fields defined in IPS_SBD_EXT.1.2.

- The evaluator checked the AGD and ensured that it provides instructions with how to configure reactions specified in IPS_SBD_EXT.1.5 for each string-based detection signature.
- The evaluator checked the AGD and ensured that it provides instructions with how rules are associated with distinct network interfaces that are capable of being associated with signatures.

The sections **'Supported test methods', 'Signature test reference' and 'Create a signature' under 'Custom Attack Definitions' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD provides the following guidance:

A Trellix IPS Custom Attack supports 2 basic classes of test:

1. **Pattern Match - Pattern matches can be used to match ASCII or binary strings. Support for common regular expression constructs is available.**
2. **Numeric Comparisons - Numeric comparisons can be used to test for a match with a given value or a range of values, depending on the numeric comparison test chosen.**

The signatures can be constructed using the guidance of each test type available for use in signatures and the required and optional parameters for the same. The AGD also provides a detailed step by step guide for customizing/creating a new signature and delegating an appropriate response for the same.

Verdict:

PASS.

5.1.8.4.6 IPS_SBD_EXT.1.3 TSS

Objective:

The evaluator shall verify that the TSS describes how the attacks defined in IPS_SBD_EXT.1.3 are processed by the TOE and what reaction is triggered when these attacks are identified.

Evaluator Findings:

The evaluator reviewed the TSS to ensure that it describes how the attacks defined in IPS_SBD_EXT.1.3 are processed by the TOE and what reaction is triggered when these attacks are identified.

The relevant information is found in the following section(s): TOE Summary Specification **IPS_SBD_EXT.1.**

Upon investigation, the evaluator found that the TSS states that**: The TSF detects the packet header-based attacks defined in IPS_SBD_EXT.1.3 by analyzing the Layer 3 and Layer 4 headers for known attack patterns. In inline mode, the TSF blocks any traffic matching one of these signatures. In all other modes, the administrator can configure how the TOE responds when it detects a header-based attack.**
The TSF detects the traffic-pattern based attacks defined in IPS_SBD_EXT.1.4 by analyzing the Layer 3 and Layer 4 headers over a period of time for known attack patterns. In inline mode, the TSF blocks any traffic matching one of these signatures. In all other modes, the administrator can configure how the TOE responds when it detects a header-based attack.

Verdict:

PASS.

5.1.8.4.7 IPS_SBD_EXT.1.3 AGD

Objective:

The evaluator shall verify that the AGD provides instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.3 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5.

Evaluator Findings:

The evaluator checked the AGD and ensured that it provides instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.3 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5.

The section **'Denial-of-Service attacks' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD provides the following information: **The threshold method provides administrators with a way to trigger alerts if a preconfigured traffic volume threshold is exceeded. The key to successfully using thresholds is to have an understanding of the normal traffic levels on the network. In most cases, an external device such as a sniffer is used to baseline the network, and the initial levels are set according to that data. Once a baseline has been established, the administrator can enable the relevant threshold for an attack and configure each with values that make sense for a particular network.**

DoS related alerts are raised when a Sensor detects volume-based DoS attacks, vulnerability based DoS attacks, and attacks by DDoS attack tools. Trellix IPS uses attack signatures to detect communication between many known DDoS attack tools, and also to detect vulnerability-based attacks. Alerts are raised in the Attack Log when such attacks are detected. In the case of volume-based attacks, Sensor looks for statistical anomalies in short-term and long-term profiles. The Sensor compares the short-term profile against the long-term profile. If there is a significant difference in the traffic levels, an alert is generated, and the Sensor blocks traffic with statistical anomalies if configured to do so.

Verdict:

PASS.

5.1.8.4.8 IPS_SBD_EXT.1.4 TSS

Objective:

The evaluator shall verify that the TSS describes how the attacks defined in IPS_SBD_EXT.1.4 are processed by the TOE and what reaction is triggered when these attacks are identified.

Evaluator Findings:

The evaluator reviewed the TSS to ensure that it describes how the attacks defined in IPS_SBD_EXT.1.4 are processed by the TOE and what reaction is triggered when these attacks are identified.

The relevant information is found in the following section(s): TOE Summary Specification **IPS_SBD_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TSF detects the packet header-based attacks defined in IPS_SBD_EXT.1.3 by analyzing the Layer 3 and Layer 4 headers for known attack patterns. In inline mode, the TSF blocks any traffic matching one of these signatures. In all other modes, the administrator can configure how the TOE responds when it detects a header-based attack.**
The TSF detects the traffic-pattern based attacks defined in IPS_SBD_EXT.1.4 by analyzing the Layer 3 and Layer 4 headers over a period of time for known attack patterns. In inline mode, the TSF blocks any traffic matching one of these signatures. In all other modes, the administrator can configure how the TOE responds when it detects a header-based attack.

Verdict:

PASS.

5.1.8.4.9 IPS_SBD_EXT.1.4 AGD

Objective:

The evaluator shall verify that the AGD provides instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.4 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5.

Evaluator Findings:

The evaluator checked the AGD and ensured that it provides instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.4 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5.

The sections **'DoS attacks defended against by Trellix IPS', 'Create custom attacks' and 'Configure custom reconnaissance attack definition' under 'Denial-of-Service attacks' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD provides **instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.4 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5. The AGD also provides a detailed step by step guide to configure a custom recon attack using the Custom Attack Editor.**

Verdict:

PASS.

5.1.8.4.10 IPS_SBD_EXT.1.5 AGD

The AGD EAs for this element are performed in conjunction with IPS_SBD_EXT.1.1, IPS_SBD_EXT.1.3, and IPS_SBD_EXT.1.4.

Verdict:

PASS.

5.1.8.4.11 IPS_SBD_EXT.1.6 AGD

Objective:

The evaluator shall verify that the AGD provides configuration instructions, if needed, to detect payload across multiple packets.

Evaluator Findings:

The evaluator checked the AGD and ensured that it provides configuration instructions, if needed, to detect payload across multiple packets.

The section **'Custom Attack Definitions' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD provides the following information: **no special configuration is required to detect payload across multiple packets.**

Verdict:

PASS.

5.2 Optional Requirements

5.2.1 Communications (FCO)

5.2.1.1 FCO_CPC_EXT.1 Component Registration Channel Definition

5.2.1.1.1 FCO_CPC_EXT.1 TSS

Objective:

- The evaluator shall examine the TSS to confirm that it:

  a) Describes the method by which a Security Administrator enables and disables communications between pairs of TOE components.

  b) Describes the relevant details according to the type of channel in the main selection made in FCO_CPC_EXT.1.2:

  • First type: the TSS identifies the relevant SFR iteration that specifies the channel used

- Second type: the TSS (with support from the AGD if selected in FTP_TRP.1.3/Join) describes details of the channel and the mechanisms that it uses (and describes how the process ensures that the key is unique to the pair of components) – see also the Evaluation Activities for FTP_TRP.1/Join.

- The evaluator shall confirm that if any aspects of the registration channel are identified as not meeting FTP_ITC.1 or FPT_ITT.1, then the ST has also selected the FTP_TRP.1/Join option in the main selection in FCO_CPC_EXT.1.2.

Evaluator Findings:

- The evaluator examined the TSS and confirmed that it:

  a) Describes the method by which a Security Administrator enables and disables communications between pairs of TOE components.

  b) Describes the relevant details according to the type of channel in the main selection made in FCO_CPC_EXT.1.2:

  - First type: the TSS identifies the relevant SFR iteration that specifies the channel used

  - Second type: the TSS (with support from the AGD if selected in FTP_TRP.1.3/Join) describes details of the channel and the mechanisms that it uses (and describes how the process ensures that the key is unique to the pair of components) – see also the Evaluation Activities for FTP_TRP.1/Join.

- The evaluator confirmed that if any aspects of the registration channel are identified as not meeting FTP_ITC.1 or FPT_ITT.1, then the ST has also selected the FTP_TRP.1/Join option in the main selection in FCO_CPC_EXT.1.2.

The relevant information is found in the following section(s): TOE Summary Specification **FCO_CPC_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **While the initial communications are being setup, the Manager is the TSF endpoint, and the Sensor is the joining component. The administrator logs in to the Sensor, specifies the Manager IP, and configures a shared secret. This shared secret is between 8 and 25 characters. The Sensor then connects to the Manager using TLS and authenticates itself using the shared secret. Both devices store the X.509 certificates, so FPT_ITT.1 TLS connections are authenticated using certificate-based TLS mutual authentication. Once the Sensor has joined the TSF, all management of the Sensor can be performed through the Manager.**

A sensor can be disabled by either issuing the 'deinstall' command from the sensor CLI or through the Device Manager in the Manager web GUI.

Verdict:

PASS.

5.2.1.1.2 FCO_CPC_EXT.1 AGD

Objective:

- The evaluator shall examine the AGD to confirm that it contains instructions for enabling and disabling communications with any individual component of a distributed TOE.
- The evaluator shall confirm that the method of disabling is such that all other components can be prevented from communicating with the component that is being removed from the TOE (preventing the remaining components from either attempting to initiate communications to the disabled component, or from responding to communications from the disabled component).
- The evaluator shall examine the AGD to confirm that it includes recovery instructions should a connection be unintentionally broken during the registration process.
- If the TOE uses a registration channel for registering components to the TOE (i.e. where the ST author uses the FTP_ITC.1/FPT_ITT.1 or FTP_TRP.1/Join channel types in the main selection for FCO_CPC_EXT.1.2) then the evaluator shall examine the Preparative Procedures to confirm that they:

a) describe the security characteristics of the registration channel (e.g. the protocol, keys and authentication data on which it is based) and shall highlight any aspects which do not meet the requirements for a steady- state inter-component channel (as in FTP_ITC.1 or FPT_ITT.1)

b) identify any dependencies between the configuration of the registration channel and the security of the subsequent inter-component communications (e.g. where AES-256 inter-component communications depend on transmitting 256 bit keys between components and therefore rely on the registration channel being configured to use an equivalent key length)

c) identify any aspects of the channel can be modified by the operational environment in order to improve the channel security and shall describe how this modification can be achieved (e.g. generating a new key pair, or replacing a default public key certificate).

- As background for the examination of the registration channel description, it is noted that the requirements above are intended to ensure that administrators can make an accurate judgement of any risks that arise from the default registration process. Examples would be the use of self-signed certificates (i.e. certificates that are not chained to an external or local Certification Authority), manufacturer-issued certificates (where control over aspects such as revocation, or which devices are issued with recognised certificates, is outside the control of the operational environment), use of generic/non-unique keys (e.g. where the same key is present on more than one instance of a device), or well-known keys (i.e. where the confidentiality of the keys is not intended to be strongly protected – note that this need not mean there is a positive action or intention to publicise the keys).
- In the case of a distributed TOE for which the ST author uses the FTP_TRP.1/Join channel type in the main selection for FCO_CPC_EXT.1.2 and the TOE relies on the operational environment to provide security for some aspects of the registration channel security then there are additional requirements on the Preparative Procedures as described in Section 3.4.1.2.

Evaluator Findings:

- The evaluator examined the AGD and confirmed that it contains instructions for enabling and disabling communications with any individual component of a distributed TOE.
- The evaluator confirmed that the method of disabling is such that all other components can be prevented from communicating with the component that is being removed from the TOE (preventing the remaining components from either attempting to initiate communications to the disabled component, or from responding to communications from the disabled component).
- The evaluator examined the AGD and confirmed that it includes recovery instructions should a connection be unintentionally broken during the registration process.
- If the TOE uses a registration channel for registering components to the TOE (i.e. where the ST author uses the FTP_ITC.1/FPT_ITT.1 or FTP_TRP.1/Join channel types in the main selection for FCO_CPC_EXT.1.2) then the evaluator examined the Preparative Procedures and confirmed that they:
    - describe the security characteristics of the registration channel (e.g. the protocol, keys and authentication data on which it is based) and highlights any aspects which do not meet the requirements for a steady- state inter-component channel (as in FTP_ITC.1 or FPT_ITT.1)
    - identify any dependencies between the configuration of the registration channel and the security of the subsequent inter-component communications (e.g. where AES-256 inter-component communications depend on transmitting 256 bit keys between components and therefore rely on the registration channel being configured to use an equivalent key length)
    - identify any aspects of the channel can be modified by the operational environment in order to improve the channel security and describes how this modification can be achieved (e.g. generating a new key pair, or replacing a default public key certificate).

- As background for the examination of the registration channel description, it is noted that the requirements above are intended and ensured that administrators can make an accurate judgement of any risks that arise from the default registration process.
- In the case of a distributed TOE for which the ST author uses the FTP_TRP.1/Join channel type in the main selection for FCO_CPC_EXT.1.2 and the TOE relies on the operational environment to provide security for some aspects of the registration channel security then there are additional requirements on the Preparative Procedures as described in Section 3.4.1.2.

The section **'Establishing Sensor-to-Manager communication' and 'Delete a Sensor from the Manager' under 'IPS Administration' of the IPS_11.1_Product_Guide and the 'Protocol features in the certified evaluated configuration' section of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD **provides information related to the method for disabling and enabling all other components within TOE. In the setup, the Manager is the TSF endpoint, and the Sensor connects to it. An admin configures the Sensor with the Manager's IP and a shared secret (8-25 characters). The Sensor then uses TLS to connect to the Manager, authenticating with the shared secret. Both devices have X.509 certificates for mutual TLS authentication (FPT_ITT.1). After joining, all Sensor management is through the Manager. Sensors can be disabled via the sensor CLI's 'deinstall' command or the Manager's web GUI. It also provides troubleshooting steps for several connection-related issues that may arise.**
The protocol features section describes the characteristics of the registration channel as "Between Sensors and the Manager, the cipher suite used to perform mutual authentication are TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384. The systems must use CA-signed RSA certificates with key size 2048 bits.". The document states that this configuration is pre-configured and fixed.

Verdict:

PASS.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_TLSC_EXT.2 Extended: TLS Client support for mutual authentication

5.2.2.1.1 FCS_TLSC_EXT.2.1 TSS

Objective:

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

Evaluator Findings:

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

The relevant information is found in the following section(s): TOE Summary Specification **FCS_TLSC_EXT.2.**

Upon investigation, the evaluator found that the TSS states that: **The TOE uses TLS with mutual authentication to provide an intra-TSF trusted channel between the TOE components. These connections are secured using TLSv1.2 with the TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ciphersuite. The TLS server compares the presented X.509 client certificate with the certificate updated through administrative configuration of certificates via the Manager GUI. If either certificate is invalid, the TOE will not establish the connection. Joining is also performed over a TLS connection. No fallback authentication functions are supported by the TSF.**

Verdict:

PASS.

5.2.2.1.2 FCS_TLSC_EXT.2.1 AGD

Objective:

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

Evaluator Findings:

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

The section **'Managing certificates for manager and sensor' of the IPS_11.1_Product_Guide and the** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD provides the following information: **The TOE employs TLS with mutual authentication to establish a secure channel between its components. TLSv1.2 with ciphersuites TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 and TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 is used. The TLS server verifies client certificates against updated X.509 certificates configured via the Manager GUI, using IPv4 addresses for reference. If certificates are invalid, connections are not established. Joining occurs over TLS without fallback authentication support.**

Verdict:

PASS.

5.2.2.2 FCS_TLSS_EXT.2 Extended: TLS Server support for mutual authentication

5.2.2.2.1 FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2 TSS

Objective:

- The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.
- The evaluator shall verify the TSS describes how the TSF uses certificates to authenticate the TLS client.
- The evaluator shall verify the TSS describes if the TSF supports any fallback authentication functions (e.g. username/password, challenge response) the TSF uses to authenticate TLS clients that do not present a certificate. If fallback authentication functions are supported, the evaluator shall verify the TSS describes whether the fallback authentication functions can be disabled.

Evaluator Findings:

- The evaluator ensured that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.
- The evaluator verified the TSS describes how the TSF uses certificates to authenticate the TLS client.
- The evaluator shall verify the TSS describes if the TSF supports any fallback authentication functions (e.g. username/password, challenge response) the TSF uses to authenticate TLS clients that do not present a certificate.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_TLSS_EXT.2.**

Upon investigation, the evaluator found that the TSS states that: **The TOE uses TLS with mutual authentication to provide an intra-TSF trusted channel between the TOE components. These connections are secured using TLSv1.2 with the TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ciphersuite. The TLS server compares the presented X.509 client certificate with the certificate updated through administrative configuration of certificates via the Manager GUI. The IPv4 address is used as the reference identifier and is compared against the SAN field, if present. Only in absence of SAN, the IPv4 address is used as the reference and compared against the CN field instead. If either certificate is invalid, the TOE will not establish the connection. Joining is also performed over a TLS connection.**

No fallback authentication functions are supported by the TSF.

Verdict:

PASS.

5.2.2.2.2 FCS_TLSS_EXT.2.3 TSS

Objective:

The evaluator shall verify that the TSS describes which types of identifiers are supported during client authentication (e.g. Fully Qualified Domain Name (FQDN)). If FQDNs are supported, the evaluator shall verify that the TSS describes that corresponding identifiers are matched according to RFC6125. For all other types of identifiers, the evaluator shall verify that the TSS describes how these identifiers are parsed from the certificate, what the expected identifiers are and how the parsed identifiers from the certificate are matched against the expected identifiers.

Evaluator Findings:

The evaluator verified that the TSS describes which types of identifiers are supported during client authentication (e.g. Fully Qualified Domain Name (FQDN)). If FQDNs are supported, the evaluator verified that the TSS describes that corresponding identifiers are matched according to RFC6125. For all other types of identifiers, the evaluator verified that the TSS describes how these identifiers are parsed from the certificate, what the expected identifiers are and how the parsed identifiers from the certificate are matched against the expected identifiers.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_TLSS_EXT.2.**

Upon investigation, the evaluator found that the TSS states that: **The TLS server compares the presented X.509 client certificate with the certificate updated through administrative configuration of certificates via the Manager GUI. The IPv4 address is used as the reference identifier and is compared against the SAN field, if present. Only in absence of SAN, the IPv4 address is used as the reference and compared against the CN field instead. If either certificate is invalid, the TOE will not establish the connection.**

Verdict:

PASS.

5.2.2.2.3 FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2 AGD

Objective:

- If the TSS indicates that mutual authentication using X.509v3 certificates is used, the evaluator shall verify that the AGD includes instructions for configuring the client-side certificates for TLS mutual authentication.
- The evaluator shall verify the AGD describes how to configure the TLS client certificate authentication function. If the TSF supports fallback authentication functions, the evaluator shall verify the AGD provides instructions for configuring the fallback authentication functions. If fallback authentication functions can be disabled, the evaluator shall verify the AGD provides instructions for disabling the fallback authentication functions.

Evaluator Findings:

- If the TSS indicates that mutual authentication using X.509v3 certificates is used, the evaluator verified that the AGD includes instructions for configuring the client-side certificates for TLS mutual authentication.
- The evaluator verified the AGD describes how to configure the TLS client certificate authentication function. If the TSF supports fallback authentication functions, the evaluator verified the AGD provides instructions for configuring the fallback authentication functions. If fallback authentication functions can be disabled, the evaluator verified the AGD provides instructions for disabling the fallback authentication functions.

The section '**Managing certificates for manager and sensor' of the IPS_11.1_Product_Guide and the** as used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the TSS states that: **The TOE uses TLS with mutual authentication to provide an intra-TSF trusted channel between the TOE components. These connections are secured using TLSv1.2 with**

the TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ciphersuite. The TLS server compares the presented X.509 client certificate with the certificate updated through administrative configuration of certificates via the Manager GUI. No fallback authentication functions are supported by the TSF.

Verdict:

PASS.

5.2.2.2.4 FCS_TLSS_EXT.2.3 AGD

Objective:

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

Evaluator Findings:

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

The section **'Managing certificates for manager and sensor' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD provides the following information: **The TLS server validates the client's X.509 certificate against the updated certificate configured via the Manager GUI. It uses the IPv4 address as the reference, checking the SAN field first and then the CN field if SAN is absent. If either certificate is invalid, the connection is not established. No fallback authentication functions are supported by the TOE.**

Verdict:

PASS.

5.2.3 Identification and Authentication (FIA)

5.2.3.1 FIA_X509_EXT.1/ITT X.509 Certificate Validation

5.2.3.1.1 FIA_X509_EXT.1/ITT X.509 TSS

Objective:

The evaluator shall examine the TSS to ensure it describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). If selected, the TSS shall describe how certificate revocation checking is performed. It is not sufficient to verify the status of a X.509 certificate only when it's loaded onto the device.

Evaluator Findings:

The evaluator examined the TSS to ensure it describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). If selected, the TSS describes how certificate revocation checking is performed. It is not sufficient to verify the status of a X.509 certificate only when it's loaded onto the device.

The relevant information is found in the following section(s): TOE Summary Specification **FIA_X509_EXT.1/ITT.**

Upon investigation, the evaluator found that the TSS states that: **The TOE uses X.509 certificates to:**

- **provide mutual authentication between different components of the TOE.**

When the TOE or TOE component receives a certificate asserting the identity of a remote system, the TOE ensures the current time is within the validity period of the certificate, the certificate has not been revoked, it contains the appropriate extendedKeyUsage purpose set (i.e., Server Authentication or Client Authentication depending on the use case), CA certificates contain the basic constraints extension with the CA flag set to TRUE and the certificate chain terminates with a trusted CA certificate.

Connections between the Sensor and Manager do not perform the revocation check. Both the Manager and Sensor portions of the TOE can generate CSRs that contain the RSA 2048 public key, Common Name, Organization, Organizational Unit, and Country. The TOE verifies that signed certificates (certificate responses) are verified to chain to a trusted CA when they are imported.

Verdict:

PASS.

5.2.3.1.2 FIA_X509_EXT.1/ITT X.509 AGD

Objective:

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

Evaluator Findings:

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

The section **'Managing certificates for manager and sensor' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD **explains the validity checks performed by the TOE for the certificates and also provides instructions for utilizing the various certificate fields such as extendedKeyUsage, basic constraints, etc. Connections between the Sensor and Manager do not perform the revocation check.**

Verdict:

PASS.

5.2.4 Protection of Security Functions (FPT)

5.2.4.1 FPT_ITT.1 Basic Internal TSF Data Transfer Protection

5.2.4.1.1 FPT_ITT.1 TSS

Objective:

- The evaluator shall examine the TSS to determine that, for all communications between components of a distributed TOE, each communications mechanism is identified in terms of the allowed protocols for that IT entity.
- The evaluator shall also confirm that all protocols listed in the TSS for these inter-component communications are specified and included in the requirements in the ST.

Evaluator Findings:

- The evaluator examined the TSS and determined that, for all communications between components of a distributed TOE, each communications mechanism is identified in terms of the allowed protocols for that IT entity.

- The evaluator also confirmed that all protocols listed in the TSS for these inter-component communications are specified and included in the requirements in the ST.

The relevant information is found in the following section(s): TOE Summary Specification **FPT_ITT.1.**

Upon investigation, the evaluator found that the TSS states that: **While the initial communications are being setup, the Manager is the TSF endpoint, and the Sensor is the joining component. The administrator logs in to the Sensor, specifies the Manager IP, and configures a shared secret. This shared secret is between 8 and 25 characters. The Sensor then connects to the Manager using TLS and authenticates itself using the shared secret. Both devices store the X.509 certificates, so FPT_ITT.1 TLS connections are authenticated using certificate-based TLS mutual authentication. Once the Sensor has joined the TSF, all management of the Sensor can be performed through the Manager.**

Verdict:

PASS.

5.2.4.1.2 FPT_ITT.1 AGD

Objective:

The evaluator shall confirm that the AGD contains instructions for establishing the relevant allowed communication channels and protocols between each pair of authorized TOE components, and that it contains recovery instructions should a connection be unintentionally broken.

Evaluator Findings:

The evaluator confirmed that the AGD contains instructions for establishing the relevant allowed communication channels and protocols between each pair of authorized TOE components, and that it contains recovery instructions should a connection be unintentionally broken.

The **IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity. The section **'Managing certificates for manager and sensor'** provides information on how to establish trust between the manager and sensor using CA signed certificates. The commands **'set manager ip'** and **'set sensor sharedsecretkey'** under section **'CLI commands'** provide instructions for configuring the manager's IP and it's shared secret key respectively.

The sub-sections **'Network connectivity'** and **'Management port configuration'** under the section titled **'Issues and status checks for the Sensor and Manager in combination'** provide recovery instructions in situations of unintentional connection outage.

Upon investigation, the evaluator found that the AGD provides **instructions for configuration of the TLS channel between the manager and sensors. The AGD includes guidelines for setting up authorized communication channels and protocols between TOE components. It also provides recovery instructions in case of unintentional connection disruptions by checking power, network connectivity, link indicator lights and cable connectivity.**

Verdict:

PASS.

5.3 Selection-Based Requirements

5.3.1 Cryptographic Support (FCS)

5.3.1.1 FCS_HTTPS_EXT.1 HTTPS Protocol

5.3.1.1.1 FCS_HTTPS_EXT.1 TSS

Objective:

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

Evaluator Findings:

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

The relevant information is found in the following section(s): TOE Summary Specification **FCS_HTTPS_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE acts as a TLS/HTTPS server to provide a web GUI to administrators. The TSF implements the server side of the HTTPS protocol in accordance with RFC 2818 by making use of a secure TLSv1.2 session to secure the HTTP session. All 'MUST' and 'REQUIRED' statements applicable to server implementations within RFC 2818 are adhered to.**

Verdict:

PASS.

5.3.1.1.2 FCS_HTTPS_EXT.1 AGD

Objective:

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

Evaluator Findings:

The evaluator examined the AGD to verify it instructs the Administrator how to configure TOE for use as an HTTPS client or HTTPS server.

Upon investigation, the evaluator found that **the 'set network'** command listed under the **'Manager Shell Commands'** section of the **Trellix IPS_11.1_Product_Guide** can be used to configure the IP address needed to access the TOE (Manager) HTTPS server and the **'CA-signed certificate for the Web Server Authentication'** section of the same guide can be used to configure the CA-signed certificate needed to use the TOE as an HTTPS server.

Verdict:

PASS.

5.3.1.2 FCS_SSHC_EXT.1 SSH Client

*5.3.1.2.1 FCS_SSHC_EXT.1.2 TSS **[TD0636 applied]***

Objective:

- The evaluator shall check to ensure that the TSS contains a list of the public key algorithms that are acceptable for use for authentication and that this list is consistent with asymetric 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.
- 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.

Evaluator Findings:

- The evaluator checked and ensured that the TSS contains a list of the public key algorithms that are acceptable for use for authentication and that this list is consistent with asymetric 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 confirmed the TSS is unambiguous in declaring the TOE's ability to authenticate itself to a remote endpoint with a user-based public key.

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

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHC_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE operates as an SSH Client with the following algorithm support:**

- **Version: v2**
- **Cipher/MAC: aes128-gcm@openssh.com, aes256-gcm@openssh.com/Implicit MAC**
- **User-based Authentication: Password, Public-key (ecdsa-sha2-nistp256)**
- **Key Exchange: ecdh-sha2-nistp256**
- **Peer Authentication (peer Server's Host key): ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256**

As an SSH client, the TOE supports both password-based authentication, and public key-based authentication with ecdsa-sha2-nistp256. The TOE is capable of generating the ECDSA based public and private keys using ECC schemes as per FIPS PUB 186-4.

The TOE is capable of identifying the peer server via the server's host key and supports the following algorithms: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256.

Verdict:

PASS.

5.3.1.2.2 FCS_SSHC_EXT.1.3 TSS

Objective:

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

Evaluator Findings:

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

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHC_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **If the TOE receives an SSH packet that exceeds 256K, the packet is dropped and logged, and the connection terminated.**

Verdict:

PASS.

5.3.1.2.3 FCS_SSHC_EXT.1.4 TSS

Objective:

- 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.

Evaluator Findings:

- The evaluator checked the description of the implementation of this protocol in the TSS and ensured that optional characteristics are specified, and the encryption algorithms supported are specified as well.
- The evaluator checked the TSS and ensured that the encryption algorithms specified are identical to those listed for this component.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHC_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE operates as an SSH Client with the following algorithm support:**

- **Version: v2**
- **Cipher/MAC: aes128-gcm@openssh.com, aes256-gcm@openssh.com/Implicit MAC**
- **User-based Authentication: Password, Public-key (ecdsa-sha2-nistp256)**
- **Key Exchange: ecdh-sha2-nistp256**
- **Peer Authentication (peer Server's Host key): ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256**

Verdict:

PASS.

*5.3.1.2.4 FCS_SSHC_EXT.1.5 TSS **[TD0636 applied]***

Objective:

- 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.
- 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.
- 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.

Evaluator Findings:

- The evaluator confirmed the TSS describes how a host-key public key (i.e., SSH server's public key) is associated with the server identity.
- The evaluator checked the description of the implementation of this protocol in the TSS and ensured that optional characteristics are specified, and the host-key public key algorithms supported by the TOE are specified as well.
- The evaluator checked the TSS and ensured that the host-key public key algorithms specified are identical to those listed for this component.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHC_EXT.1.**

Upon investigation, the evaluator found that the TSS of the **ST identifies the host-key public key algorithms supported for SSH connections by the TOE. The following public key algorithms are identified as supported within the TSS,**

- **ecdsa-sha2-nistp256**
- **ssh-rsa**
- **rsa-sha2-256**
- **rsa-sha2-512**

The SSH Client functionality is used by a Sensor to provide a trusted channel with the SCP server, where it associates a host-key public key with a server identity by using a 'known_hosts' file on its filesystem.

- TSF does not claims the 'x509v3-based public key authentication' algorithms.

Verdict:

PASS.

5.3.1.2.5 FCS_SSHC_EXT.1.6 TSS

Objective:

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.

Evaluator Findings:

The evaluator checked the TSS and ensured that it lists the supported data integrity algorithms, and that the list corresponds to the list in this component.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHC_EXT.1.**

Upon investigation, the evaluator found that the TSS the **of ST identifies the data integrity algorithms supported for SSH connections by the TOE. The following data integrity algorithms are identified as supported within the TSS,**

- **Implicit MAC**

The data integrity algorithms specified in the definition of the SFR are consistent with the description within the TSS of ST.

Verdict:

PASS.

5.3.1.2.6 FCS_SSHC_EXT.1.7 TSS

Objective:

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.

Evaluator Findings:

The evaluator checked the TSS and ensured that it lists the supported key exchange algorithms, and that the list corresponds to the list in this component.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHC_EXT.1.**

Upon investigation, the evaluator found that the TSS of the **ST identifies the key exchange algorithms supported for SSH connections by the TOE. The following key exchange algorithms are identified as supported within the TSS,**

- **ecdh-sha2-nistp256**

The key exchange algorithms specified in the definition of the SFR are consistent with the description within the TSS of ST.

Verdict:

PASS.

5.3.1.2.7 FCS_SSHC_EXT.1.8 TSS

Objective:

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.

Evaluator Findings:

The evaluator checked 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.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHC_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE checks for both time-based as well as volume-based threshold configurations and rekeying is performed on the basis of whichever threshold is reached first. If a rekey is initiated by the remote server/client, the TOE also honours such an attempt.**

Verdict
PASS.

*5.3.1.2.8 FCS_SSHC_EXT.1.2 AGD **[TD0636 applied]***

Objective:

The evaluator shall check the AGD 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.

Evaluator Findings:

The SSH connections initiated by the TOE can only utilize the claimed mechanisms. The TOE does not require configuration since it is pre-configured and fixed and these mechanisms cannot be modified.

Verdict:

PASS.

5.3.1.2.9 FCS_SSHC_EXT.1.4 AGD

Objective:

The evaluator shall also check the AGD and ensured 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).

Evaluator Findings:

The evaluator also checked the AGD and ensured 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).

Upon investigation, the evaluator found that **the TOE does not require configuration since it is pre-configured and fixed and these mechanisms cannot be modified. The information about the default configurations can be found in the section 'Sensor features in FIPS compliant images' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide.**

Verdict:

PASS.

5.3.1.2.10 FCS_SSHC_EXT.1.5 AGD

Objective:

The evaluator shall also check the AGD 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).

Evaluator Findings:

The evaluator also checked the AGD 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).

Upon investigation, the evaluator found that **the TOE does not require configuration since it is pre-configured and fixed and these mechanisms cannot be modified. The information about the default configurations can be found in the section 'Sensor features in FIPS compliant images' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide.**

Verdict:

PASS.

5.3.1.2.11 FCS_SSHC_EXT.1.6 AGD

Objective:

The evaluator shall also check the AGD and ensured 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).

Evaluator Findings:

The evaluator also checked the AGD and ensured 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).

Upon investigation, the evaluator found that **the TOE does not require configuration since end users lack shell access and cannot modify these mechanisms and the "none" MAC algorithm is not allowed as part of the default configuration. The section 'Sensor features in FIPS compliant images' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide provides information about the MAC algorithms being used by the TOE.**

Verdict:

PASS.

5.3.1.2.12 FCS_SSHC_EXT.1.7 AGD

Objective:

The evaluator shall also check the AGD 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.

Evaluator Findings:

The evaluator also checked the AGD and ensured 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.

Upon investigation, the evaluator found that **the TOE does not require configuration since it is pre-configured and fixed and these mechanisms cannot be modified. The information about the default configurations can be found in the section 'Sensor features in FIPS compliant images' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide.**

Verdict:

PASS.

5.3.1.2.13 FCS_SSHC_EXT.1.8 AGD

Objective:

- If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, then the evaluator shall check that the AGD describes how to configure those thresholds. Either the allowed values are specified in the AGD 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 AGD describes that the TOE reacts to the first threshold reached.

Evaluator Findings:

- If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, then the evaluator checked that the AGD describes how to configure those thresholds. Either the allowed values are specified in the AGD 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 checked that the AGD describes that the TOE reacts to the first threshold reached.

The section **'Reconfiguration of SSH and SSHD' of the Trellix Intrusion Prevention System Manager Appliance Product Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD provides the following information: **The TOE does not require configuration since those are pre-configured and fixed and these mechanisms cannot be modified. The rekey threshold can be changed via shell access only. The section 'Reconfiguration of SSH and SSHD' of the Trellix Intrusion Prevention System Manager Appliance Product Guide provides information related to the default rekey thresholds based on volume and time limit.**

Verdict:

PASS.

5.3.1.3 FCS_SSHS_EXT.1 SSH Server

*5.3.1.3.1 FCS_SSHS_EXT.1.2 TSS **[TD0631 applied]***

Objective:

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

Evaluator Findings:

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

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHS_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE operates as an SSH Server with the following algorithm support:**

- **Version: v2**
- **Cipher/MAC: aes128-gcm@openssh.com, aes256-gcm@openssh.com/Implicit MAC**

- **Hostkey: ecdsa-sha2-nistp256**
- **Key Exchange: ecdh-sha2-nistp256**
- **User Authentication: Password, Public-key (ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256)**

The TOE supports password-based authentication by looking up the username in /etc/passwd and comparing the hash of the password to the value in /etc/shadow. If the credentials correspond to an entry in the files, the user is successfully authenticated and is authorized to access the TOE. Public key-based authentication implemented by the TOE succeeds if the matching private key is used. This is verified by confirming that the presented private key corresponds to the public key associated with the user in the 'authorized_keys' file on the TOE filesystem.
As an SSH client, the TOE supports both password-based and public key-based authentication with the above listed algorithms' support.

Verdict:

PASS.

5.3.1.3.2 FCS_SSHS_EXT.1.3 TSS

Objective:

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

Evaluator Findings:

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

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHS_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **If the TOE receives an SSH packet that exceeds 256K, the packet is dropped and logged, and the connection terminated.**

Verdict:

PASS.

5.3.1.3.3 FCS_SSHS_EXT.1.4 TSS

Objective:

- 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.

Evaluator Findings:

- The evaluator checked the description of the implementation of this protocol in the TSS and ensured that optional characteristics are specified, and the encryption algorithms supported are specified as well.
- The evaluator checked the TSS and ensured that the encryption algorithms specified are identical to those listed for this component.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHS_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE operates as an SSH Server with the following algorithm support:**

- **Version: v2**
- **Cipher/MAC: aes128-gcm@openssh.com, aes256-gcm@openssh.com/Implicit MAC**
- **Hostkey: ecdsa-sha2-nistp256**
- **Key Exchange: ecdh-sha2-nistp256**
- **User Authentication: Password, Public-key (ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256)**

Verdict:

PASS.

*5.3.1.3.4 FCS_SSHS_EXT.1.5 TSS **[TD0631 applied]***

Objective:

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.

Evaluator Findings:

The evaluator checked the description of the implementation of this protocol in the TSS and ensured that the SSH server's host public key algorithms supported are specified and that they are identical to those listed for this component.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHS_EXT.1.**

Upon investigation, the evaluator found that the TSS **ST identifies the host-key public key algorithms supported for SSH connections by the TOE. The following public key algorithms are identified as supported within the TSS,**

- **ecdsa-sha2-nistp256**

Verdict:

PASS.

5.3.1.3.5 FCS_SSHS_EXT.1.6 TSS

Objective:

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.

Evaluator Findings:

The evaluator checked the TSS and ensured that it lists the supported data integrity algorithms, and that the list corresponds to the list in this component.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHS_EXT.1.**

Upon investigation, the evaluator found that the TSS of the **of ST identifies the data integrity algorithms supported for SSH connections by the TOE. The following data integrity algorithms are identified as supported within the TSS,**

- **Implicit MAC**

The data integrity algorithms specified in the definition of the SFR are consistent with the description within the TSS of ST.

Verdict:

PASS.

5.3.1.3.6 FCS_SSHS_EXT.1.7 TSS

Objective:

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.

Evaluator Findings:

The evaluator checked the TSS and ensured that it lists the supported key exchange algorithms, and that the list corresponds to the list in this component.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHS_EXT.1.**

Upon investigation, the evaluator found that the TSS **ST identifies the key exchange algorithms supported for SSH connections by the TOE. The following key exchange algorithms are identified as supported within the TSS,**

- **ecdh-sha2-nistp256**

The key exchange algorithms specified in the definition of the SFR are consistent with the description within the TSS of ST.

Verdict:
PASS.

5.3.1.3.7 FCS_SSHS_EXT.1.8 TSS

Objective:
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.

Evaluator Findings:
The evaluator checked 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.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_SSHS_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE checks for both time-based as well as volume-based threshold configurations and rekeying is performed on the basis of whichever threshold is reached first. If a rekey is initiated by the remote server/client, the TOE also honours such an attempt.**

Verdict
PASS.

5.3.1.3.8 FCS_SSHS_EXT.1.4 AGD

Objective:
The evaluator shall also check the AGD 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).

Evaluator Findings:
The evaluator also checked the AGD and ensured 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).

Upon investigation, the evaluator found that **the TOE does not require configuration since it is pre-configured and fixed and these mechanisms cannot be modified. The information about the default configurations can be found in the section 'Sensor features in FIPS compliant images' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide.**

Verdict:
PASS.

5.3.1.3.9 FCS_SSHS_EXT.1.5 AGD

Objective:

The evaluator shall also check the AGD 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).

Evaluator Findings:

The evaluator also checked the AGD and ensured 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).

Upon investigation, the evaluator found that **the TOE does not require configuration since it is pre-configured and fixed and these mechanisms cannot be modified. The information about the default configurations can be found in the section 'Sensor features in FIPS compliant images' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide.**

Verdict:

PASS.

5.3.1.3.10 FCS_SSHS_EXT.1.6 AGD

Objective:

The evaluator shall also check the AGD 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).

Evaluator Findings:

Upon investigation, the evaluator found that **the TOE does not require configuration since end users lack shell access and cannot modify these mechanisms and the "none" MAC algorithm is not allowed as part of the default configuration. The section 'Sensor features in FIPS compliant images' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide provides information about the MAC algorithms being used by the TOE.**

Verdict:

PASS.

5.3.1.3.11 FCS_SSHS_EXT.1.7 AGD

Objective:

The evaluator shall also check the AGD 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.

Evaluator Findings:

The evaluator also checked the AGD and ensured 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.

Upon investigation, the evaluator found that **the TOE does not require configuration since it is pre-configured and fixed and these mechanisms cannot be modified. The information about the default configurations can be found in the section 'Sensor features in FIPS compliant images' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide.**

Verdict:

PASS.

5.3.1.3.12 FCS_SSHS_EXT.1.8 AGD

Objective:

- If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, then the evaluator shall check that the AGD describes how to configure those thresholds. Either the allowed values are specified in the AGD 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 AGD describes that the TOE reacts to the first threshold reached.

Evaluator Findings:

- If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, then the evaluator checked that the AGD describes how to configure those thresholds. Either the allowed values are specified in the AGD 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 checked that the AGD describes that the TOE reacts to the first threshold reached.

The section **'Reconfiguration of SSH and SSHD' of the Trellix Intrusion Prevention System Manager Appliance Product Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD provides the following information: **The TOE does not require configuration since end users lack shell access and cannot modify these mechanisms. The rekey threshold can be changed via shell access only. The section 'Reconfiguration of SSH and SSHD' of the Trellix Intrusion Prevention System Manager Appliance Product Guide provides information related to the default rekey thresholds based on volume and time limit.**

Verdict:

PASS.

5.3.1.4 FCS_TLSC_EXT.1 Extended: TLS Client Protocol Without Mutual Authentication

5.3.1.4.1 FCS_TLSC_EXT.1.1 TSS

Objective:

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

Evaluator Findings:

- The evaluator checked the description of the implementation of this protocol in the TSS and ensured that the ciphersuites supported are specified.
- The evaluator checked the TSS and ensured that the ciphersuites specified include those listed for this component.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_TLSC_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE (Manager) operates as TLS Client to provide a trusted channel with the syslog server. The TOE provides the following version and algorithm support:**

- **TLS Version: v1.2**
- **Supported ciphersuites:**
  - **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256**
  - **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384**

Verdict:

PASS.

5.3.1.4.2 FCS_TLSC_EXT.1.2 TSS

Objective:

- The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the administrator/application- configured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.
- If IP addresses are supported in the CN as reference identifiers, the evaluator shall ensure that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order.
- The evaluator shall also ensure that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced.

Evaluator Findings:

- The evaluator ensured that the TSS describes the client's method of establishing all reference identifiers from the administrator/application- configured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.
- If IP addresses are supported in the CN as reference identifiers, the evaluator ensured that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order.
- The evaluator also ensured that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_TLSC_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE automatically parses the reference identifier from the connection parameters, using the FQDN or IPv4 address as the reference identifier. When validating the server certificate, the TSF matches the configured reference identifier against the DNS or IPv4 SAN fields in the presented certificate (if present) and falls back to the CN if the SAN is not present. Wildcards are supported in the leftmost label of the FQDN SAN field but not in the CN field. An IPv4 reference identifier in the CN field is converted to its corresponding binary representation in network byte order and has canonical format enforced in accordance with RFC 3986. The TOE does not establish a trusted channel if the server certificate is invalid and does not support any administrative override mechanism.**

Verdict:

PASS.

5.3.1.4.3 FCS_TLSC_EXT.1.4 TSS

Objective:

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

Evaluator Findings:

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

The relevant information is found in the following section(s): TOE Summary Specification **FCS_TLSC_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE presents the Supported Elliptic Curves Extension indicating support for P-256 and P-384 in the Client Hello.**

Verdict:

PASS.

5.3.1.4.4 FCS_TLSC_EXT.1.1 AGD

Objective:

The evaluator shall check the AGD to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

Evaluator Findings:

The evaluator checked the AGD section **'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide and examined that no additional configuration is required for enforcing the opted ciphersuites. The information about the default ciphersuites can be found in the same section of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide.**

Verdict:

PASS.

5.3.1.4.5 FCS_TLSC_EXT.1.2 AGD

Objective:

- The evaluator shall ensure that the AGD describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator shall ensure that the AGD provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.
- Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1, the SFR selects attributes from RFC 5280, and FCO_CPC_EXT.1.2 selects "no channel"; the evaluator shall verify the AGD provides instructions for establishing unique reference identifiers based on RFC5280 attributes.

Evaluator Findings:

- The evaluator ensured that the AGD describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator ensured that the AGD provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.
- Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1, the SFR selects attributes from RFC 5280, and FCO_CPC_EXT.1.2 selects "no channel"; the evaluator verified the AGD provides instructions for establishing unique reference identifiers based on RFC5280 attributes.

The section **'Configure a Syslog server' of the IPS_11.1_Product_Guide and the 'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD activity states that: **The reference identifier is the IP address or fully qualified domain name of the configured endpoint (matching the type used to configure the endpoint) and may be found in the SAN or CN fields of the presented certificate.**

Verdict:

PASS.

5.3.1.4.6 FCS_TLSC_EXT.1.4 AGD

Objective:

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

Evaluator Findings:

If the TSS indicates that the Supported Elliptic Curves/Supported Groups Extension must be configured to meet the requirement, the evaluator verified that the AGD includes configuration of the Supported Elliptic Curves/Supported Groups Extension.

The section **'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found **the TOE does not require configuration since it is pre-configured and fixed and these mechanisms cannot be modified. The information about the default configurations for the Supported Elliptic Curves/Supported Groups Extension can be found in the section 'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide.**

Verdict:

PASS.

5.3.1.5 FCS_TLSS_EXT.1 Extended: TLS Server Protocol Without Mutual Authentication

5.3.1.5.1 FCS_TLSS_EXT.1.1 TSS

Objective:

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

Evaluator Findings:

- The evaluator checked the description of the implementation of this protocol in the TSS and ensured that the ciphersuites supported are specified.
- The evaluator checked the TSS and ensured that the ciphersuites specified are identical to those listed for this component.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_TLSS_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE acts as a TLS/HTTPS server to provide a web GUI to administrators. This server supports TLSv1.2 using the following ciphersuites, with all other TLS/SSL versions being rejected:**

- **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256**
- **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384**

Verdict:

PASS.

5.3.1.5.2 FCS_TLSS_EXT.1.2 TSS

Objective:

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

Evaluator Findings:

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

The relevant information is found in the following section(s): TOE Summary Specification **FCS_TLSS_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **This server supports TLSv1.2 using the following ciphersuites, with all other TLS/SSL versions being rejected:**

- **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256**
- **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384**

Verdict:

PASS.

*5.3.1.5.3 FCS_TLSS_EXT.1.3 TSS **[TD0635 applied]***

Objective:

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

Evaluator Findings:

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

The relevant information is found in the following section(s): TOE Summary Specification **FCS_TLSS_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **The TOE acts as a TLS/HTTPS server to provide a web GUI to administrators. This server supports TLSv1.2 using the following ciphersuites, with all other TLS/SSL versions being rejected:**

- **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256**
- **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384**

Key Establishment is performed using Elliptic Curve Diffie-Hellman P-256 keys.

Verdict:

PASS.

*5.3.1.5.4 FCS_TLSS_EXT.1.4 TSS **[TD0569 applied]***

Objective:

- The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246) and/or if session resumption based on session tickets is supported (RFC 5077).
- If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption.
- The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets .

- If session tickets are supported, the evaluator shall verify that the TSS describes that session tickets adhere to the structural format provided in Section 4 of RFC 5077 and if not, a justification shall be given of the actual session ticket format.
- If the TOE claims a (D)TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator shall verify that the TSS describes how session resumption operates (i.e. what would trigger a full handshake, e.g. checking session status, checking Session ID, etc.). If multiple contexts are used the TSS describes how session resumption is coordinated across those contexts. In case session establishment and session resumption are always using a separate context, the TSS shall describe how the contexts interact with respect to session resumption (in particular regarding the session ID). It is acceptable for sessions established in one context to be resumable in another context.

Evaluator Findings:

- The evaluator verified that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246) and/or if session resumption based on session tickets is supported (RFC 5077).
- TSF does not support session resumption based on session tickets.
- Key lengths and algorithms used to protect session tickets are not applicable.
- Session tickets are not supported by the TSF.
- If the TOE claims a (D)TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator verified that the TSS describes how session resumption operates (i.e. what would trigger a full handshake, e.g. checking session status, checking Session ID, etc.). If multiple contexts are used the TSS describes how session resumption is coordinated across those contexts. In case session establishment and session resumption are always using a separate context, the TSS describes how the contexts interact with respect to session resumption (in particular regarding the session ID). It is acceptable for sessions established in one context to be resumable in another context.

The relevant information is found in the following section(s): TOE Summary Specification **FCS_TLSS_EXT.1.**

Upon investigation, the evaluator found that the TSS states that: **If the session ID belongs to a previously valid/successful session, the TOE reuses the same session ID and hence resumes the session, following a shorter, partial TLS handshake. However, in case a session ID belonging to a previously invalid/failed TLS session is presented, the TOE implicitly rejects it by presenting a new session ID in the 'Server Hello' message, and proceeds with a fresh and complete handshake, thereby not resuming the previous session. The TOE supports session resumption using session IDs, in accordance with RFC 5246.**

Verdict:

PASS.

5.3.1.5.5 FCS_TLSS_EXT.1.1 AGD

Objective:

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

Evaluator Findings:

The evaluator checked the AGD and ensured that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).

The section **'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the **TOE's TLS behavior is pre-configured and fixed and no additional configuration is required for enforcing the opted ciphersuites. The information about the default ciphersuites can be found in the section 'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide.**

Verdict:

PASS.

5.3.1.5.6 FCS_TLSS_EXT.1.2 AGD

Objective:

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

Evaluator Findings:

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

The section **'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the **AGD identifies the method for configuring TLS server communications on the TOE. Only TLS v1.2 is supported by the TOE.**

Verdict:

PASS.

5.3.1.5.7 FCS_TLSS_EXT.1.3 AGD

Objective:

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

Evaluator Findings:

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

The section **'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the **TOE's TLS behavior is pre-configured and fixed, and no additional configuration is required for enforcing the opted ciphersuites. The information about the default ciphersuites can be found in the section 'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide. Only TLS v1.2 is supported by the TOE.**

Verdict:

PASS.

*5.3.1.5.8 FCS_TLSS_EXT.1.4 AGD **[TD0569 applied]***

Objective:

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

Evaluator Findings:

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

The section **'Protocol features in the certified evaluated configuration' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD activity states that: **The Target of Evaluation (TOE) supports session resumption using session ID, which does not require any separate configuration.**

Verdict:

PASS.

5.3.2 Identification and Authentication (FIA)

5.3.2.1 FIA_X509_EXT.1/Rev X.509 Certificate Validation

5.3.2.1.1 FIA_X509_EXT.1/Rev TSS

Objective:

- The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). It is expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected). It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).
- The TSS shall describe when revocation checking is performed and on what certificates. If the revocation checking during authentication is handled differently depending on whether a full certificate chain or only a leaf certificate is being presented, any differences must be summarized in the TSS section and explained in the AGD.

Evaluator Findings:

- The evaluator ensured the TSS describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). It is expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected). It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).
- The TSS describes when revocation checking is performed and on what certificates. If the revocation checking during authentication is handled differently depending on whether a full certificate chain or only a leaf certificate is being presented, any differences must be summarized in the TSS section and explained in the AGD.

The relevant information is found in the following section(s): TOE Summary Specification **FIA_X509_EXT.1/Rev.**

Upon investigation, the evaluator found that the TSS states that: **When the TOE or TOE component receives a certificate asserting the identity of a remote system, the TOE ensures the current time is within the validity period of the certificate, the certificate has not been revoked, it contains the appropriate extendedKeyUsage purpose set (i.e., Server Authentication or Client Authentication depending on the use case), CA certificates contain the basic constraints extension with the CA flag set to TRUE and the certificate chain terminates with a trusted CA certificate.**

Revocation check is performed on the leaf and intermediate CA certificates via OCSP at the time of loading as well as at the time of connection establishment. There is no difference in handling of revocation checking during authentication irrespective of whether a full certificate chain or only a leaf certificate is being presented. If the TOE cannot establish a connection to the OCSP server, the TOE will reject the certificate. This action is not administrator configurable. Hence, the OCSP server must be configured for the TOE to be able to consume the certificates.

Connections between the Sensor and Manager do not perform the revocation check.

Verdict:

PASS.

5.3.2.1.2 FIA_X509_EXT.1/Rev AGD

Objective:

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

Evaluator Findings:

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

The sections **'Considerations for CA-signed certificate chain' and 'OCSP guidelines' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the section **'Considerations for CA-signed certificate chain' of the AGD states that: The certificate must comply with the following parameters:**
• ExtendedKeyUsage: TLS WebServerAuthentication and TLS WebClientAuthentication
• KeyUSage: Must not be set to Critical.

The section **'OCSP guidelines' of the AGD states that: Validity check of syslog server certificates is performed on the TOE during upload as well as during session establishment with the syslog server. 1. OCSP responders must be setup for the leaf as well as intermediate CA certificates so that the TOE can process the certificates with OCSP URLs in them. Manager Administration 170 Trellix Intrusion Prevention System 11.1.x Product Guide 2. If the connection cannot be established for the validity check, the administrator should check that:**
• The OCSP responders are setup for all leaf and intermediate CA certificates during loading as well as connection establishment.
• OCSP responders are set up as per the information in the OCSP URL of the certificates such as IP address and port number
• Index file being passed to the OCSP responders include correct details of all certificates being verified.
• Appropriate OCSP signer and CA certificates and private keys are passed to the responders.
OCSP requests and responses use CertID.issuerNameHash and CertID.issuerKeyHash parameters to validate the revocation status of CA certificates.

Verdict:
PASS.

5.3.2.2 FIA_X509_EXT.2 X.509 Certificate Authentication

5.3.2.2.1 FIA_X509_EXT.2 TSS

Objective:

- The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the AGD for configuring the operating environment so that the TOE can use the certificates.
- The evaluator shall examine the TSS and confirmed that it describes the behaviour of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel.
- The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the AGD contains instructions on how this configuration action is performed.

Evaluator Findings:

- The evaluator checked the TSS and ensured that it describes how the TOE chooses which certificates to use, and any necessary instructions in the AGD for configuring the operating environment so that the TOE can use the certificates.
- The evaluator examined the TSS and confirmed that it describes the behaviour of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel.
- The evaluator verified that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator ensured that the AGD contains instructions on how this configuration action is performed.

The relevant information is found in the following section(s): TOE Summary Specification **FIA_X509_EXT.2.**

Upon investigation, the evaluator found that the TSS states that: **The TOE uses X.509 certificates to:**

- **provide mutual authentication between different components of the TOE**
- **verify the identity of the Syslog server**
- **identify the TOE to administrators connecting to the web GUI**

For all three of the above-mentioned functionalities, the TOE uses the certificate chain loaded and configured under the corresponding section of the Manager GUI.

When the TOE or TOE component receives a certificate asserting the identity of a remote system, the TOE ensures the current time is within the validity period of the certificate, the certificate has not been revoked, it contains the appropriate extendedKeyUsage purpose set (i.e., Server Authentication or Client Authentication depending on the use case), CA certificates contain the basic constraints extension with the CA flag set to TRUE and the certificate chain terminates with a trusted CA certificate.

Revocation check is performed on the leaf and intermediate CA certificates via OCSP at the time of loading as well as at the time of connection establishment. There is no difference in handling of revocation checking during authentication irrespective of whether a full certificate chain or only a leaf certificate is being presented. If the TOE cannot establish a connection to the OCSP server, the TOE will reject the certificate. This action is not administrator configurable. Hence, the OCSP server must be configured for the TOE to be able to consume the certificates.

Connections between the Sensor and Manager do not perform the revocation check.

Moreover, the evaluator checked instructions in the AGD for configuring the operating environment so that the TOE can use the certificates. The sections **'Configure a Syslog server' , 'Managing certificates for manager and sensor' and 'OCSP guidelines' of the IPS_11.1_Product_Guide** were used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD **provides instructions for configuring the operating environment so that the TOE can use the certificates.**

Verdict:

PASS.

5.3.2.2.2 FIA_X509_EXT.2 AGD

Objective:

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

Evaluator Findings:

The evaluator also ensured that the AGD describes the configuration required in the operating environment so the TOE can use the certificates. The AGD also includes any required configuration on the TOE to use the certificates. The AGD

also describes the steps for the Security Administrator to follow if the connection cannot be established during the validity check of a certificate used in establishing a trusted channel.

The sections **'Managing certificates for manager and sensor', 'OCSP guidelines' and 'Validation errors while importing the CA-signed certificate chain' of the IPS_11.1_Product_Guide** were used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD **outlines the necessary configuration in the operating environment for the TOE to utilize certificates. It also specifies any required configuration on the TOE itself for certificate usage. Additionally, the AGD provides instructions for the Security Administrator in case a connection cannot be established during the certificate validity check for establishing a trusted channel.**

Verdict:

PASS.

5.3.2.3 FIA_X509_EXT.3 Extended: X509 Certificate Requests

5.3.2.3.1 FIA_X509_EXT.3 TSS

Objective:

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

Evaluator Findings:

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

The relevant information is found in the following section(s): TOE Summary Specification **FIA_X509_EXT.3.**

Upon investigation, the evaluator found that **device-specific information is not selected by the ST author.**

Verdict:

PASS.

5.3.2.3.2 FIA_X509_EXT.3 AGD

Objective:

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

Evaluator Findings:

The evaluator checked and ensured that the AGD contains instructions on requesting certificates from a CA, including generation of a Certificate Request. If the ST author selects "Common Name", "Organization", "Organizational Unit", or "Country", the evaluator ensured that the AGD includes instructions for establishing these fields before creating the Certification Request.

The section **'Managing certificates for manager and sensor' of the IPS_11.1_Product_Guide** was used to determine the verdict of this assurance activity.

Upon investigation, the evaluator found that the AGD provides **instructions which include the complete set of steps necessary to configure a fully formed CSR containing each of the fields described in FIA_X509_EXT.3. The AGD also provides instructions for generating CSRs from the GUI.**

Verdict:

PASS.

6 Security Assurance Requirements

6.1 TOE Summary Specification (ASE_TSS.1)

6.1.1 ASE_TSS.1.1C

Objective:

The evaluator shall examine the TSS to determine that it is clear which TOE components contribute to each SFR or how the TSFI that is identified as being security relevant.

The evaluator shall verify the sufficiency to fulfil the related SFRs. This includes checking that the TOE as a whole fully covers all SFRs and that all functionality that is required to be audited is in fact audited regardless of the component that carries it out.

Evaluator Findings:

The evaluator examined the TSS to determine that it is clear which TOE components contribute to each SFR or how the TSFI that is identified as being security relevant.

The evaluator verified the sufficiency to fulfil the related SFRs. This includes checking that the TOE as a whole fully covers all SFRs and that all functionality that is required to be audited is in fact audited regardless of the component that carries it out.

Verdict:

PASS.

6.2 Basic Functional Specification (ADV_FSP)

6.2.1 ADV_FSP.1

6.2.1.1 ADV_FSP.1-1

Objective:

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.

Evaluator Findings:

The evaluator examined the AGD (interface documentation) to verify that it describes the purpose and method of use for each TSFI that is identified as being security relevant. The evaluator examined the entire AGD. The evaluator verified the AGD describes the purpose and method of use for each security relevant TSFI by verifying the AGD satisfies all of the AGD Evaluation Activities.

Verdict:

PASS.

6.2.1.2 ADV_FSP.1-2

Objective:

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.

Evaluator Findings:

The evaluator examined the AGD (interface documentation) to develop a mapping of the interfaces to SFRs. The evaluator examined the entire AGD. Each AGD Activity is associated with a specific SFR. The Evaluation Findings for each AGD Activity identify the relevant interfaces, thus providing a mapping.

Verdict:

PASS.

6.2.1.3 ADV_FSP.1-3

Objective:

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.

Evaluator Findings:

The evaluator examined the AGD (interface documentation) to verify that it identifies and describes the parameters for each TSFI that is identified as being security relevant. The evaluator examined the entire AGD. The evaluator verified the AGD describes the parameters for each security relevant TSFI by verifying the AGD satisfies all of the AGD Activities.

Verdict:

PASS.

6.2.1.4 ADV_FSP.1-5

Objective:

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

Evaluator Findings:

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

Verdict:

PASS.

6.3 Operational User Guidance (AGD_OPE)

6.3.1 AGD_OPE.1

6.3.1.1 AGD_OPE.1-1

Objective:

The evaluator shall ensure the AGD 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.

Evaluator Findings:

The evaluator checked the requirements below are met by the AGD. The AGD is distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration. Upon investigation, the evaluator found that the CC guidance will be published with the CC certificate on www.niap-ccevs.org.

Verdict:

PASS.

6.3.1.2 AGD_OPE.1-2

Objective:

The evaluator shall ensure that the AGD 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.

Evaluator Findings:

The evaluator ensured that the AGD is provided for every Operational Environment that the product supports as claimed in the Security Target. The section titled **'An overview of Trellix Intrusion Prevention System' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide and the section 'Hardware specifications' of the Trellix Intrusion Prevention System Manager Appliance Product Guide** was used to determine the verdict of this assurance activity. The AGD lists the following platforms claimed in the Security Target:

Trellix Intrusion Prevention System Sensor Appliances:

- **IPS-NS9500**
- **IPS-NS7600**
- **IPS-NS7500**
- **IPS-NS3600**
- **IPS-NS3200**

Trellix Intrusion Prevention System Manager Appliance:

- **NSM-MAPL-NG (XEON SILVER 4210)**
- **NSM-MAPL -NG (XEON SILVER 4114)**

Verdict:

PASS.

6.3.1.3 AGD_OPE.1-3

Objective:

The evaluator shall ensure that the AGD 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.

Evaluator Findings:

The evaluator ensured that the AGD contains instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. While performing the AGD Activities for the cryptographic SFRs, the evaluator ensured that the AGD contains the necessary instructions for configuring the cryptographic engines.

Verdict:

PASS.

6.3.1.4 AGD_OPE.1-4

Objective:

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

Evaluator Findings:

The entire AGD was used to determine the verdict of this work unit. Each confirmation command indicates tested options. Additionally, the sections titled **'Product Functionality not Included in the Scope of the Evaluation'** and **'Security Functions Provided by the TOE'** make it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs along with the ones out of testing scope.

Verdict:

PASS.

6.3.1.5 AGD_OPE.1-5

Objective:

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

The AGD 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.

The AGD 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:

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).

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.

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

Evaluator Findings:

The evaluator verified the AGD contains instructions for configuring any cryptographic engines in AGD_OPE.1 Test #3.

The evaluator verified the AGD describes the process for verifying updates in FPT_TUD_EXT.1 AGD activity.

The evaluator verified the AGD makes it clear which security functionality is covered by the Evaluation Activities in AGD_OPE.1 Test #4.

Verdict:

PASS.

6.4 Preparative Procedures (AGD_PRE)

6.4.1 AGD_PRE.1

6.4.1.1 AGD_PRE.1-1

Objective:

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).

Evaluator Findings:

The evaluator examined the Preparative procedures to ensure they include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality. The evaluator reviewed the sections titled **'Configure a Syslog server'** of the **IPS_11.1_Product_Guide**. The evaluator found that the following objectives listed in the Security Target describe how the Operational Environment must meet:

OE.PHYSICAL

OE.NO_GENERAL_PURPOSE

OE.NO_THRU_TRAFFIC_PROTECTION

OE.TRUSTED_ADMN

OE.UPDATES

OE.ADMIN_CREDENTIALS_SECURE

OE.RESIDUAL_INFORMATION

OE.CONNECTIONS (IPS)

Verdict:

PASS.

6.4.1.2 AGD_PRE.1-2

Objective:

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.

Evaluator Findings:

The evaluator checked the requirements below are met by the preparative procedures. The entire AGD was used to determine the verdict of this work unit. Upon investigation, the evaluator found that the AGD describes each of the devices in the operating environment, including,

Local Management Console

Remote Management Workstation

External IT systems

Update Server

Syslog Server

OCSP Server

Sensors

The section titled **'An overview of Trellix Intrusion Prevention System' of the Trellix Intrusion Prevention System 11.1 FIPS and CC Certification Guide and the section 'Hardware specifications' of the Trellix Intrusion Prevention System Manager Appliance Product Guide** of AGD identifies the following supported platform:

NS9500

NS3200

NS3600

NS7500

NS7600

NSM-MAPL-NG (XEON SILVER 4210)

NSM-MAPL -NG (XEON SILVER 4114)

Verdict:

PASS.

6.4.1.3 AGD_PRE.1-3

Objective:

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

Evaluator Findings:

The evaluator checked the requirements are met by the preparative procedures. The entire AGD was used to determine the verdict of this work unit. Upon investigation, the evaluator found that AGD describes all of the functions necessary to install and configure the TOE to work in the target operating environment, including:

Local Management Console

Remote Management Workstation

External IT systems

Update Server

Syslog Server

Verdict:

PASS.

6.4.1.4 AGD_PRE.1-4

Objective:

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.

Evaluator Findings:

The evaluator ensured the preparative procedures include instructions to manage the security of the TSF as a product and as a component of the larger operational environment. The entire AGD was used to determine the verdict of this work unit. The same commands, configurations, and interfaces used to install the TOE are also used for ongoing management, so this is satisfied by AGD_PRE.1-3.

Verdict:

PASS.

6.4.1.5 AGD_PRE.1-5

Objective:

In addition, the evaluator shall ensure that the following requirements are also met. The preparative procedures must:

include instructions to provide a protected administrative capability; and

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

Evaluator Findings:

The evaluator ensured the preparative procedures include instructions to provide a protected administrative capability and changing default passwords. The **command 'set password' under 'Manager Shell Commands' and section 'Configure password complexity settings' under 'Manager Administration' of the IPS_11.1_Product_Guide** were used to determine the verdict of this work unit. The AGD describes changing the default password associated with the root account and configuring SSH for remote administration.

Verdict:

PASS.

6.5 Assurance Activities (ALC)

6.5.1 ALC_CMC.1

Objective:

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.

Evaluator Findings:

The evaluator verified that the ST, TOE and AGD are all labeled with the same hardware versions and software. The information is specific enough to procure the TOE and it includes hardware models and software versions. The evaluator

checked the TOE software version and hardware identifiers during testing by examining the actual machines used for testing.

Verdict:

PASS.

6.5.2 ALC_CMS.1

Objective:

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

Evaluator Findings:

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

Verdict:

PASS.

6.6 Independent Testing – Conformance (ATE_IND)

6.6.1 ATE_IND.1

Objective:

The evaluator performs the CEM work units associated with the ATE_IND.1 SAR. Specific testing requirements and EAs are captured for each SFR in Sections 2, 3 and 4. The evaluator should consult Appendix 709 when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

Evaluator Findings:

The evaluator examined the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST. Upon investigation, the evaluator found that each instance of the TOE used in testing was consistent with TOE description found in the Security Target. Additionally, the evaluator found that the TOE version is consistent with what was specified in the Security Target. The evaluator examined the TOE to determine that it has been installed properly and is in a known state. The details of the installed TOE and any configuration performed with the TOE are found in the separate Test Reports. The evaluator prepared a test plan that covers all of the testing actions for ATE_IND.1 in the CEM and in the SFR-related Evaluation Activities.

Verdict:

PASS.

6.7 Vulnerability Survey (AVA_VAN)

6.7.1 AVA_VAN.1

6.7.1.1 AVA_VAN.1-1 **[TD0564 applied] [Labgram #116]**

Objective:

The evaluator shall document their analysis and testing of potential vulnerabilities with respect to this requirement.

Evaluator Findings:

The evaluator documented their analysis and testing of potential vulnerabilities with respect to this requirement. Public searches were performed against all keywords found within the Security Target and AGD that may be applicable to specific TOE components. This included protocols, TOE software version, and TOE hardware to ensure sufficient

coverage under AVA. The evaluator searched the Internet for potential vulnerabilities in the TOE using the web sites listed below. The sources of the publicly available information are provided below:

- https://nvd.nist.gov/view/vuln.search
- http://cve.mitre.org/cve
- https://www.cvedetails.com/vulnerability-search.php
- https://www.kb.cert.org/vuls/search/
- www.exploitsearch.net
- www.securiteam.com
- http://nessus.org/plugins/index.php?view=search
- http://www.zerodayinitiative.com/advisories
- https://www.exploit-db.com
- https://www.rapid7.com/db/vulnerabilities

The evaluator performed the public domain vulnerability searches using the following key words. The search was performed on May 6th, 2024.

- Trellix Intrusion Prevention System 11.1.
- Trellix IPS Manager.
- Trellix IPS Sensor.
- Trellix NS3200.
- Trellix NS3600
- Trellix NS7500.
- Trellix NS7600.
- Trellix NS9500.
- Intel XEON GOLD 6230.
- Intel XEON GOLD 5218N.
- Intel ATOM C2538.
- Intel XEON SILVER 4210.
- Intel XEON SILVER 4114.
- Intel XEON SILVER 4416+
- Intel XEON D-1734NT
- MLOS 3.9.
- McAfee Linux Operating System 3.9.
- Apache Tomcat 9.0.85
- MariaDB 10.6.16
- BouncyCastle 2.2.0
- OpenSSL 1.0.2zh
- OpenSSL-fips 2.0.5
- OpenSSH 7.4P1-33
- OpenSSH 7.8p1
- Rsyslogd 8.24.0-57

The evaluation lab examined each result provided from NVD and Exploit Search to determine if the current TOE version or component within the environment was vulnerable. Based upon the analysis, any issues found that were generated were patched in the TOE version and prior versions, mitigating the risk factor.

Verdict:

PASS.

6.7.1.2 AVA_VAN.1-2

Objective:

The evaluator shall perform the following activities to generate type 4 flaw hypotheses:

Fuzz testing

Examine effects of sending:

mutated packets carrying each ‘Type’ and ‘Code’ value that is undefined in the relevant RFC for each of ICMPv4 (RFC 792) and ICMPv6 (RFC 4443)

mutated packets carrying each ‘Transport Layer Protocol’ value that is undefined in the respective RFC for IPv4 (RFC 791) IPv6 (RFC 2460) should also be covered if it is supported and claimed by the TOE.

Since none of these packets will belong to an allowed session, the packets should not be processed by the TOE, and the TOE should not be adversely affected by this traffic. Any results that are unexpected (e.g., core dumps) are candidates for a flaw hypothesis.

Mutation fuzz testing of the remaining fields in the required protocol headers. This testing requires sending mutations of well- formed packets that have both carefully chosen and random values inserted into each header field in turn (i.e. testing is to include both carefully chosen and random insertion test cases). The original well-formed packets would be accepted as part of a normal existing communication stream and may still be accepted as valid packets when subject to the carefully chosen mutations (the individual packet alone would be valid although its contents may not be valid in the context of preceding and/or following packets), but will often not be valid packets when random values are inserted into fields. The carefully chosen values should include semantically significant values that can be determined from the type of the data that the field represents, such as values indicating positive and negative integers, boundary conditions, invalid binary combinations (e.g. for flag sets with dependencies between bits), and missing start or end values. Randomly chosen values may not result in well-formed packets but are included nonetheless to see whether they can lead to the device entering an insecure state. Any results that are unexpected (e.g., core dumps) are candidates for a flaw hypothesis.

Evaluator Findings:

The evaluator documented the fuzz testing results with respect to this requirement. The evaluation lab examined each result from fuzz testing to determine if the TOE improperly processes packets. Based upon the analysis, no unexpected results occurred. Therefore, no Type 4 hypotheses were generated.

Verdict:

PASS.

7 Detailed Test Cases (Test Activities)

7.1 Sensor

7.1.1 Audit

7.1.1.1 FAU_GEN.1 Test #1

Item	Data
Test Assurance Activity	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. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.
Test Steps	• Trigger each auditable event on the TOE. Verify that each audit record is generated and contains the required information
Expected Test Results	• **The TOE accurately generates audit records for all the required auditable events.** • **Evidence- Snapshot showing generated logs for audit records.**
Pass/Fail with Explanation	Pass. The audit records associated with each test case are recorded with each test case. A comparison of required audit records to the presented audit records was additionally performed. This analysis shows that each required audit record is generated by the TOE.

7.1.1.2 FPT_STM_EXT.1 Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• **Verify the current time on the TOE.** • **Set a new time on the manager and let it sync with the TOE.** • **Verify that the time on the TOE was updated.** • **Verify logs were generated for the time change.**
Expected Test Results	• **Logs successfully reflect changed time according to the set time on TOE.** • **TOE generates logs for the time change.**
Pass/Fail with Explanation	Pass. Observed that Security Admin is able to modify time on TOE.

7.1.1.3 FPT_STM_EXT.1 Test #2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	NA. TOE does not support the use of an NTP server.

7.1.1.4 FPT_STM_EXT.1 Test #3

Item	Data
Test Assurance Activity	[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.
Pass/Fail with Explanation	NA. TOE is not a vND.

7.1.1.5 FTP_ITC.1 Test #1

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FTP_ITC.1 Test #4.

7.1.1.6 FTP_ITC.1 Test #2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FTP_ITC.1 Test #4.

7.1.1.7 FTP_ITC.1 Test #3

Item	Data
Test Assurance Activity	The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FTP_ITC.1 Test #4. The SSH connections were maintained or re-established as necessary, and all data sent between the system was encrypted. This satisfies the testing requirements.

7.1.1.8 FTP_ITC.1 Test #4

Item	Data
Test Assurance Activity	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: 1. A duration that exceeds the TOE's application layer timeout setting, 2. 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.
Test Steps	• Initiate an SSH connection with the update server. • Disconnect the cable between the TOE and the update server for less than the application layer timeout and verify via PCAPs that the SSH session was not dropped, and that all data continues to be encrypted. The existing SSH connection was maintained, and all data sent between the systems is encrypted. • Disconnect the cable between the TOE and the update server for more than the application layer timeout. Verify the "packet_write_wait" error in the SSH connection and reinitiate the SSH connection. • Verify via PCAPs that the SSH session was re-established, and that all data is encrypted. The existing SSH connection was dropped, and a new connection was established before any user data was transmitted.
Expected Test Results	• **The TOE should send encrypted data before and after the temporary disconnection with the external IT entity.**
Pass/Fail with Explanation	Pass. The SSH connections were maintained or re-established as necessary, and all data sent between the system was encrypted. This satisfies the testing requirements.

7.1.2 Crypto

7.1.2.1 FCS_CKM.2 DH14

This test was removed by TD0580.

7.1.2.2 FCS_CKM.1 RSA

Item	Data
Test Assurance Activity	Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed

	automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). **Key Generation for FIPS PUB 186-4 RSA Schemes** The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent *e*, the private prime factors *p* and *q*, the public modulus *n* and the calculation of the private signature exponent d. Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include: a) Random Primes: • Provable primes • Probable primes b) Primes with Conditions: • Primes p1, p2, q1, q2, p and q shall all be provable primes • Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes • Primes p1, p2, q1, q2, p and q shall all be probable primes To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation.
Pass/Fail with Explanation	Algorithm: RSA KeyGen Key size / Modulus: 2048 CAVP #: A3350 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.3 FCS_CKM.1 ECC

Item	Data
Test Assurance Activity	Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). **Key Generation for Elliptic Curve Cryptography (ECC)** FIPS 186-4 ECC Key Generation Test For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the

	evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation. FIPS 186-4 Public Key Verification (PKV) Test For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.
Pass/Fail with Explanation	Algorithm: ECDSA KeyGen, ECDSA KeyVer Curves: P-256, P-384 CAVP #: A3350 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.4 FCS_CKM.2 SP800-56A

Item	Data

Test Assurance Activity	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 The evaluator shall verify a TOE's implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag. Function Test The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested. The evaluator shall obtain the DKM, the corresponding TOE's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields. If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret. The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values. If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm. Validity Test The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOE's public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE id fields.

	The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOE's static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass). The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results with the results using a known good implementation verifying that the TOE detects these errors.
Pass/Fail with Explanation	Algorithm: KAS-ECC-SSC Sp800-56Ar3 Curves: P-256, P-384 CAVP #: A3350 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.5 FCS_COP.1/DataEncryption AES-GCM

Item	Data
Test Assurance Activity	AES-GCM Test The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths: 128 bit and 256 bit keys a) Two plaintext lengths. One of the plaintext lengths shall be a nonzero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported. a) Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported. b) Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested. The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known. The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.
Pass/Fail with Explanation	Algorithm: AES GCM Key size: 128, 256 CAVP #: A3350 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.6 FCS_COP.1/SigGen ECDSA

Item	Data
Test Assurance Activity	**ECDSA Algorithm Tests** **ECDSA FIPS 186-4 Signature Generation Test** For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator sha long messages and obtain for each message a public key and the resulting signature values R and S. T the evaluator shall use the signature verification function of a known good implementation. **ECDSA FIPS 186-4 Signature Verification Test** For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator sha 1024-bit message, public key and signature tuples and modify one of the values (message, public key 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Pass/Fail with Explanation	Algorithm: ECDSA SigGen, ECDSA SigVer Curves: P-256 CAVP #: A3350 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.7 FCS_COP.1/SigGen RSA

Item	Data
Test Assurance Activity	RSA Signature Algorithm Tests Signature Generation Test The evaluator generates or obtains 10 messages for each modulus size/SHA combination supported by the TOE. The TOE generates and returns the corresponding signatures. The evaluator shall verify the correctness of the TOE's signature using a trusted reference implementation of the signature verification algorithm and the associated public keys to verify the signatures. Signature Verification Test For each modulus size/hash algorithm selected, the evaluator generates a modulus and three associated key pairs, (d, e). Each private key d is used to sign six pseudorandom messages each of 1024 bits using a trusted reference implementation of the signature generation algorithm. Some of the public keys, e, messages, or signatures are altered so that signature verification should fail. For both the set of original messages and the set of altered messages: the modulus, hash algorithm, public key e values, messages, and signatures are forwarded to the TOE, which then attempts to verify the signatures and returns the verification results. The evaluator verifies that the TOE confirms correct signatures on the original messages and detects the errors introduced in the altered messages.
Pass/Fail with Explanation	Algorithm: RSA SigGen Key size / Modulus: 2048 CAVP #: A3350 Algorithm: RSA SigVer Key size / Modulus: 2048 CAVP #: A3350, A3353 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.8 FCS_COP.1/Hash

Item	Data
Test Assurance Activity	The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented testmacs. The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP. Short Messages Test - Bit-oriented Mode The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. Short Messages Test - Byte-oriented Mode The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. Selected Long Messages Test - Bit-oriented Mode The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. Selected Long Messages Test - Byte-oriented Mode The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. Pseudorandomly Generated Messages Test This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

Pass/Fail with Explanation	Algorithm: SHA-1, SHA-256, SHA-384, SHA-512 CAVP #: A3350 Algorithm: SHA-256, SHA-384 CAVP #: A3353 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.9 FCS_COP.1/KeyedHash

Item	Data
Test Assurance Activity	For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and message data using a known good implementation.
Pass/Fail with Explanation	Algorithm: HMAC (SHA-256, SHA-384, SHA-512) CAVP #: A3350 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.10 FCS_RBG_EXT.1

Item	Data
Test Assurance Activity	The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A). If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call. The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator. Entropy input: the length of the entropy input value must equal the seed length.

	Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length. Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied. Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.
Pass/Fail with Explanation	Algorithm: CTR DRBG Mode: AES-256 CAVP #: A3350 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.3 Auth

7.1.3.1 FIA_AFL.1 Test #1

Item	Data
Test Assurance Activity	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. ***TD0570 is applied.***
Test Steps	• **Set the threshold for maximum number of authentication attempts.** • **Attempt to connect to the TOE with incorrect credentials until a lockout is triggered.** • **Verify via logs that the user account is now locked out.** • **Attempt to login using the correct credentials and verify that it fails due to the lockout.** • **Verify the failed login attempt via logs.**
Expected Test Results	**The maximum number of successive unsuccessful attempts can be configured on the TOE. The TOE does not allow for access to the device even with correct credentials after an account fails authentication successively for the configured maximum number of unsuccessful attempts.**
Pass/Fail with Explanation	Pass. An authentication failure disallows user from connecting to the TOE after the configured number of incorrect attempts.

7.1.3.2 FIA_AFL.1 Test #2a

Item	Data
Test Assurance Activity	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 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). ***TD0570 is applied.***
Test Steps	• **Set the threshold for maximum number of authentication attempts.** • **Attempt to connect to the TOE with incorrect credentials until a lockout is triggered.** • **Verify via logs that the user account is now locked out.** • **Attempt to login using the correct credentials and verify that it fails due to the lockout.** • **Manually unlock the user account.** • **Verify via logs that the user account is unlocked.** • **Login with good credentials and verify that it succeeds.** • **Verify the successful login with logs.**
Expected Test Results	• **After the Administrator unlocks a locked account, the user is able to successfully login over SSH using the correct username and password. The audit logs show the administrator unlocking the locked account, followed by a successful login by the user.**
Pass/Fail with Explanation	Pass. The TOE successfully rejects log in with valid credentials till the locked account is unlocked by a Security Administrator.

7.1.3.3 FIA_AFL.1 Test #2b

Item	Data
Test Assurance Activity	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 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows: 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.
Pass/Fail with Explanation	NA. Time period selection is not made for the sensor.

7.1.3.4 FIA_PMG_EXT.1 Test #1

Item	Data
Test Assurance Activity	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.

Test Steps	• Set the minimum password requirements:
	  ○ Minimum 15 character length.
	  ○ Minimum 2 upper case.
	  ○ Minimum 2 lower case.
	  ○ Minimum 2 digit.
	  ○ Minimum 2 special character.
	• Attempt to create 15 characters password with username: good & password: AB1CD7E!a@bc1de.
	• Attempt to create 15 characters password with username: good1 & password: FG2HI8J#f$gh2ij.
	• Attempt to create 15 characters password with username: good2 & password: KL3MN9O%k^lm3no.
	• Attempt to create 15 characters password with username: good3 & password: PQ4RS0T&p*qr4st.
	• Attempt to create 15 characters password with username: good4 & password: UV5WX1Y(u)vw5xy.
	• Attempt to create 15 characters password with username: good5 & password: ZA6BC2D!z@ab6cd.
	• Verify all the users are created on the TOE.
	• Verify with the logs that users 'good, good1, good2, good3, good4, good5' are created.

Expected Test Results	**The TOE accepts valid password combinations that meet the requirements. Audit logs show that the user with the valid password combination has been added successfully.**
Pass/Fail with Explanation	Pass. The TOE successfully creates user accounts with the configured password strength. This meets the testing requirements.

7.1.3.5 FIA_PMG_EXT.1 Test #2

Item	Data
Test Assurance Activity	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.

Test Steps	• Set the minimum password requirements:
	  ○ Minimum 15 character length.
	  ○ Minimum 2 upper case.
	  ○ Minimum 2 lower case.
	  ○ Minimum 2 digit.
	  ○ Minimum 2 special character.
	
	• Attempt to create a user with a missing upper case character in the password with username: bad & password: ab1cd7e!a@bc1de
	• Confirm that the password could not be assigned for the newly created user via logs.
	
	• Attempt to create a user with missing lower-case character in password with username: bad1 & password: FG2HI8J#F$GH2IJ
	• Confirm that the password could not be assigned for the newly created user via logs.
	
	• Attempt to create a user with missing digits in the password with username: bad2 & password: KLmMNra%k^lmsno
	• Confirm that the password could not be assigned for the newly created user via logs.
	
	• Attempt to create a user with a missing special character in the password with username: bad3 & password: PQ4RS0T2prqr4st
	• Confirm that the password could not be assigned for the newly created user via logs.
	
	• Attempt to create a user with less than 15 characters in password username: bad4 & password: UV5WX1Y(u)vw
	• Confirm that the password could not be assigned for the newly created user via logs.

Expected Test Results	**The TOE only accepts valid password combinations. Audit logs show that addition of users with bad password combinations result in failure due to Invalid Password.**
Pass/Fail with Explanation	Pass. User accounts password cannot be set without configured password requirements being met. This meets the testing requirements.

7.1.3.6 FIA_UIA_EXT.1 Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	SSH: • Attempt to login from a remote CLI with valid username and invalid password. • Confirm that access was denied with logs. • Log into the TOE using invalid username and valid password and verify the failure. • Verify the reason for failure via logs. • Log into the TOE using an invalid username and invalid password and verify the failure. • Verify the reason for failure via logs. • Log into the TOE using correct credentials and verify the successful connection. • Verify the logs reflecting successful a connection. • Log into the TOE using public-key based authentication. • Verify the logs reflecting successful a connection. • Log into the TOE using public-key based authentication with incorrect public key. • Verify the reason for failure via logs. CONSOLE: • Verify audit log status on the TOE. • Verify console logs event status on the TOE. • Attempt to login from local console with valid username and invalid password. • Confirm that access was denied with logs. • Log into the TOE using invalid username and valid password and verify the failure. • Verify the reason for failure via logs. • Log into the TOE using an invalid username and invalid password and verify the failure. • Verify the reason for failure via logs. • Log into the TOE using correct credentials and verify the successful connection. • Verify the logs reflecting successful a connection.
Expected Test Results	• **The TOE should allow the user with correct credentials and reject the user with incorrect credentials.** • **TOE should generate logs for the successful and unsuccessful login attempts.**
Pass/Fail with Explanation	Pass. Presenting incorrect authentication credentials results in denial of access to the TOE. Presenting correct authentication credentials results in access being allowed to the TOE. This meets the testing requirements.

7.1.3.7 FIA_UIA_EXT.1 Test #2

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: Test 2: The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.
Test Steps	• Verify the warning banner is displayed prior to authentication. SSH: • Enter commands such as ping and cd as the username and use correct password and verify the unsuccessful login. • Verify the failure via logs. • Enter commands such as ping and cd as the password and use correct username and verify the unsuccessful login. • Verify the failure via logs. • Enter commands such as ping and cd as the username and use public-key based authentication. • Verify the failure via logs. Console: • Before logging in via local console, attempt to execute authenticated commands such as show, show auditlog, and show status. This will fail. • Verify the failure via logs. • Enter commands such as ping and cd as the password and use correct username and verify the unsuccessful login. • Verify the failure via logs.
Expected Test Results	• **The TOE should not expose services to an unauthenticated remote entity, and it should only display the login banner.** • **Evidence – Snap showing only display banner is present before login.**
Pass/Fail with Explanation	Pass. No system services except the login banner is available to an unauthenticated user connecting remotely. This meets the testing requirements

7.1.3.8 FIA_UIA_EXT.1 Test #3

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FIA_UIA_EXT.1_Test#2. No services except displaying a banner are available to the local administrator attempting to login to the TOE via the console.

7.1.3.9 FIA_UIA_EXT.1 Test #4

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.
Pass/Fail with Explanation	Pass. The Sensor performs the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2. Therefore, no additional testing is required for this activity. This meets the testing requirements

7.1.3.10 FIA_UAU.7 Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• **Connect to the TOE via console and enter incorrect login credentials and verify that most obscured feedback is provided by the TOE.** • **Verify the failure with logs.** • **Connect to the TOE via console with correct login credentials and verify that most obscured feedback is provided by the TOE.** • **Verify the logs reflecting successful login.**
Expected Test Results	**The TOE should not provide anything other than obscured feedback, i.e 'Incorrect Password' when entered credentials are incorrect and no feedback with correct credentials while entering authenticating information.**
Pass/Fail with Explanation	Pass. The TOE only provides obscured feedback when using incorrect credentials and provided nothing when using correct credentials.

7.1.3.11 FMT_MOF.1/ManualUpdate Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	SSH • Create a user without Security Administrator privileges. • Login as the newly created user. • Attempt to update the device. This will fail as the required options are unavailable. • Verify via logs that the attempt was unsuccessful. Console • Login as the newly created user. • Attempt to update the device. This will fail as the required options are unavailable. • Verify via logs that the attempt was unsuccessful.

Expected Test Results	**The users with low privileges must not be able to update the sensor. Only the security administrator can issue the update command.**
Pass/Fail with Explanation	Pass. Update functions are blocked for users with low privileges.

7.1.3.12 FMT_MOF.1/ManualUpdate Test #2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. This test has been completed as part of the requirements specified in FPT_TUD_EXT.1 Test#1. This meets the testing requirements.

7.1.3.13 FMT_SMF.1 Test #1

Item	Data
Test Assurance Activity	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. **TD0631 has been applied.**

Test Steps	The TSF shall be capable of performing the following management functions:
	• Ability to administer the TOE locally and remotely;
	• Ability to configure the access banner;
	• Ability to configure the session inactivity time before session termination or locking;
	• Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;
	• Ability to configure the authentication failure parameters for FIA_AFL.1;
	• [
	  ○ Ability to configure the cryptographic functionality;
	  ○ Ability to configure the interaction between TOE components;
	  ○ Ability to re-enable an Administrator account;
	  ○ Ability to set the time which is used for time-stamps;
	  ○ Ability to import X.509v3 certificates to the TOE's trust store;
	  ○ Ability to manage the trusted public keys database;]
	• Enable, disable signatures applied to sensor interfaces, and determine the behavior of IPS functionality
	• Modify these parameters that define the network traffic to be collected and analyzed:
	  ○ Source IP addresses (host address and network address)
	  ○ Destination IP addresses (host address and network address)
	  ○ Source port (TCP and UDP)
	  ○ Destination port (TCP and UDP)
	  ○ Protocol (IPv4 and IPv6)
	  ○ ICMP type and code
	• Update (import) signatures
	• Create custom signatures
	• Configure anomaly detection

	• Enable and disable actions to be taken when signature or anomaly matches are detected • Modify thresholds that trigger IPS reactions • Modify the duration of traffic blocking actions • Modify the known-good and known-bad lists (of IP addresses or address ranges) • Configure the known-good and known-bad lists to override signature-based IPS policies
Expected Test Results	**All management functions identified in Security Target should be met by presenting correct test cases.**
Pass/Fail with Explanation	Pass. Throughout the various security functionality testing of the TOE, FMT_SMF.1 Specification of Management Functions requirements have been met.

7.1.3.14 FMT_SMR.2 Test #1

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. There are three interfaces where these can be tested [console/SSH/TLS (channel between manager and sensor)], and all test cases use these interfaces. The evaluator has met this requirement through execution of the entirety of this test report by performing actions via all three interfaces.

7.1.3.15 FTA_SSL.3 Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• **Configure a SSH inactive time out period of 2 minutes on the TOE.** • **Login to TOE and record the current time.** • **Let the connection be idle for the configured time limit and verify that the session terminates.** • **Verify the connection termination via logs.** • **Configure a SSH inactive time out period of 5 minutes on the TOE.** • **Login to TOE and record the current time.** • **Let the connection be idle for the configured time limit and verify that the session terminates.** • **Verify the connection termination via logs.**
Expected Test Results	• **The TOE should support configuration for several different values for the inactivity time period and successfully terminate the session after the timeout period.** • **TOE should generate logs for session timeout**
Pass/Fail with Explanation	Pass. The TOE terminated the idle session after the configured time limit is crossed.

7.1.3.16 FTA_SSL.4 Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• **Log onto the TOE through a local console interface.** • **Perform some activity.** • **Using the instructions provided by the user guide, log off from the TOE.** • **Verify the logs for the session.**
Expected Test Results	• **The TOE should terminate the local session after the user logs off.** • **TOE should generate logs for the session termination.**
Pass/Fail with Explanation	Pass. The TOE terminates the local connection when the user issues the exit command.

7.1.3.17 FTA_SSL.4 Test #2

Item	Data
Test Assurance Activity	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.
Test Steps	• **Log onto the TOE using SSH.** • **Perform some activity.** • **Using the instructions provided by the user guide, log off from the TOE.** • **Verify the logs for the session.**
Expected Test Results	• **The TOE should terminate the remote session after the user logs off.** • **TOE should generate logs for the session termination.**
Pass/Fail with Explanation	Pass. The TOE terminates the remote session when the user issues the exit command.

7.1.3.18 FTA_SSL_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	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 reauthentication is needed when trying to unlock the session.
Test Steps	• **Set the local console time out period to 1 minute (60 seconds).** • **Connect to the TOE from the local console.** • **Verify the login time and remain idle for the configured time.** • **Verify that the session is terminated with logs.** • **Set the local console time out period to 2 minutes (120 seconds).**

	• **Connect to the TOE from the local console.** • **Verify the login time and remain idle for the configured time.** • **Verify that the session is terminated with logs.**
Expected Test Results	• **The TOE should terminate the session after the configured time period.** • **TOE logs should show session termination.**
Pass/Fail with Explanation	Pass. The TOE terminated the session for the user on local console after inactivity time limit is crossed. This meets the testing requirements.

7.1.3.19 FTA_TAB.1 Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• **Using the manager GUI, configure the login banner for the TOE.** • **Login via SSH and verify that the configured login banner is displayed.** • **Login via local console and verify that the configured login banner is displayed.**
Expected Test Results	• **The TOE should support the display of the configured login banner.** • **Evidence - screenshot showing the configured login banners.**
Pass/Fail with Explanation	Pass. A login banner can be configured for all the methods that can be used to access the TOE. This meets the testing requirements.

7.1.3.20 FTP_TRP.1/Admin Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• **Connect to the TOE using SSH.** • **Verify the successful connection using packet capture. Also, verify that the traffic is encrypted.** • **Verify the connection via logs.**
Expected Test Results	• **The TOE must encrypt the traffic for the connection.** • **Evidence – Packet capture showing successful connection.**
Pass/Fail with Explanation	Pass. The remote user is successfully able to access the TOE via TLS/SSH connection. This meets the testing requirements.

7.1.3.21 FTP_TRP.1/Admin Test #2

Item	Data
Test Assurance Activity	The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FTP_TRP.1/Admin Test#1. The TOE does not transmit any data in plaintext and only sends encrypted traffic.

7.1.4 Distributed

7.1.4.1 FAU_GEN.1 Test #2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. The audit records associated with each test case are recorded with each test case, in the test report for each component. A comparison of required audit records to the presented audit records was additionally performed and is included in section 6. This analysis shows that each required audit record is generated by each TOE component according to the allocation of SFRs, meeting the test requirements.

7.1.4.2 FAU_GEN.2 Test #1

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FCO_CPC_EXT.1 Test#1.1.

7.1.4.3 FAU_STG_EXT.5 Test #1

Item	Data
Test Assurance Activity	For each type of TOE component, the evaluator shall perform a representative subset of auditable actions and ensure that these actions cause the generation of appropriately formed audit records. Generation of such records can be observed directly on the distributed TOE component (if there is appropriate interface), or indirectly after transmission to a central location.
Test Steps	• Verify that channel between sensor and manager is established. • Configure the syslog server on the manger and verify the connection being established. • Perform actions on the sensor to generate audit records. • Verify the audit data being sent from the TOE to the manager. • Verify that the traffic between the manager and the TOE is encrypted via packet capture. • Verify the audit data being sent from the manager to the configured syslog server. • Verify that traffic between syslog and manager is encrypted through packet capture.
Expected Test Results	**Audit records generated by the sensor component are transmitted to the Manager component and then also to the syslog server.**
Pass/Fail with Explanation	Pass. The sensor is able to generate records and securely send it to the syslog server via the manager. This meets the testing requirements.

7.1.4.4 FAU_STG_EXT.5 Test #3

Item	Data
Test Assurance Activity	Test 3: For each type of TOE component that, in the evaluated configuration, is capable of transmitting audit information to another TOE component (as specified in FTP_ITT.1 or FTP_ITC.1, respectively), the evaluator shall configure a secure channel and confirm that audit records generated as a result of actions taken by the evaluator are securely transmitted. It is sufficient to observe negotiation and establishment of the secure channel with the TOE component and the subsequent transmission of encrypted data to confirm this functionality. Alternatively, the following steps shall be performed: The evaluator induces audit record transmission, then reviews the packet capture around the time of transmission and verifies that no audit data is transmitted in the clear.
Pass/Fail with Explanation	Pass. This test case is exercised in FAU_STG_EXT.5 Test #1 where a secure connection between sensor and manager was observed. This meets the testing requirement.

7.1.4.5 FCO_CPC_EXT.1 Test #1.1

Item	Data
Test Assurance Activity	Test 1.1: the evaluator shall confirm that an IT entity that is not currently a member of the distributed TOE cannot communicate with any component of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE components that it is required to communicate with (non-equivalent TOE components are as defined in the minimum configuration for the distributed TOE)
Test Steps	• Check the device manager in the manager GUI for configured sensors. • Configure the sensor to communicate with the manager without adding it to the manager configuration. • Verify that the sensor does not get added as a configured device in the device manager. • Verify the trust establishment failure via the manager logs.

	• Verify via packet capture that the sensor attempts trust establishment with the manager and fails to do so. • Configure the manager to communicate with the sensor by adding the latter in the device manager. • Configure the corresponding sensor to communicate with the manager. • Verify that the sensor gets added as a configured device in the device manager. • Verify the trust establishment success via the manager logs. • Verify via packet capture that the configured sensor attempts trust establishment with the manager and succeeds.
Expected Test Results	• **The TOE cannot communicate with any component of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE components.**
Pass/Fail with Explanation	Pass. The TOE cannot communicate with any component of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE components.

7.1.4.6 FCO_CPC_EXT.1 Test #1.2

Item	Data
Test Assurance Activity	Test 1.2: the evaluator shall confirm that after enablement, an IT entity can communicate only with the components that it has been enabled for. This includes testing that the enabled communication is successful for the enabled component pair, and that communication remains unsuccessful with any other component for which communication has not been explicitly enabled Some TOEs may set up the registration channel before the enablement step is carried out, but in such a case the channel must not allow communications until after the enablement step has been completed.
Test Steps	• Verify the status of the enabled sensor in the device manager GUI. • Verify via logs that the sensor is enabled successfully. • Verify via packet capture the successful communication between the manager and the enabled sensor. • While this sensor is enabled, initiate communication from another non-enabled sensor. • Verify that the non-enabled sensor does not get added as a configured device in the device manager. • Verify the trust establishment failure via the manager logs. • Verify via packet capture that the non-enabled sensor attempts trust establishment with the manager and fails to do so.
Expected Test Results	• **TOE communication with an enabled component is successful.** • **TOE communication with a component that hasn't been explicitly enabled is unsuccessful.**
Pass/Fail with Explanation	Pass. A sensor cannot communicate with the manager until the sensor is enabled by a Security Administrator.

7.1.4.7 FCO_CPC_EXT.1 Test #2

Item	Data

Test Assurance Activity	Test 2: The evaluator shall separately disable each TOE component in turn and ensure that the other TOE components cannot then communicate with the disabled component, whether by attempting to initiate communications with the disabled component or by responding to communication attempts from the disabled component.
Test Steps	• Verify the status of the trust channel of the TOE with the manager. • Disable the TOE's trust with the manager and verify the disconnected status. • Verify the trust channel termination via the sensor logs. • Verify via packet capture that the TOE terminates the connection with the manager and does not reattempt to establish a connection with the manager. • Verify that the TOE does not re-establish a connection with the manager.
Expected Test Results	**Disabled components are unable to successfully communicate with the TOE.**
Pass/Fail with Explanation	Pass. Disabled components are unable to successfully communicate with the TOE when disabled or disconnected.

7.1.4.8 FCO_CPC_EXT.1 Test #3

Item	Data
Test Assurance Activity	Test 3: The evaluator shall carry out the following tests according to those that apply to the values of the main (outer) selection made in the ST for FCO_CPC_EXT.1.2. 1) If the ST uses the first type of communication channel in the selection in FCO_CPC_EXT.1.2 then the evaluator tests the channel via the Evaluation Activities for FTP_ITC.1 or FPT_ITT.1 according to the second selection – the evaluator shall ensure that the test coverage for these SFRs includes their use in the registration process. 2) If the ST uses the second type of communication channel in the selection in FCO_CPC_EXT.1.2 then the evaluator tests the channel via the Evaluation Activities for FTP_TRP.1/Join. 3) If the ST uses the 'no channel' selection, then no test is required.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FPT_ITT.1. The TOE cannot communicate with any component of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE components. This meets the testing requirements.

7.1.4.9 FCO_CPC_EXT.1 Test #4

Item	Data
Test Assurance Activity	Test 4: The evaluator shall perform one of the following tests, according to the TOE characteristics identified in its TSS and operational guidance: 1) If the registration channel is not subsequently used for intercomponent communication, and in all cases where the second selection in FCO_CPC_EXT.1.2 is made (i.e. using FTP_TRP.1/Join) then the evaluator shall confirm that the registration channel can no longer be used after the registration process has completed, by attempting to use the channel to communicate with each of the endpoints after registration has completed 2) If the registration channel is subsequently used for intercomponent communication then the evaluator shall confirm that any aspects identified in the operational guidance as necessary to meet the requirements for a steady-state intercomponent channel (as in FTP_ITC.1 or FPT_ITT.1) can indeed be carried out (e.g. there might be a requirement to replace the default key pair and/or public key certificate).

Pass/Fail with Explanation	Pass. Test covered by FCO_CPC_EXT.1 Test #1.1 (the registration channel is subsequently used for intercomponent communication).

7.1.4.10 FCO_CPC_EXT.1 Test #5

Item	Data
Test Assurance Activity	For each aspect of the security of the registration channel that operational guidance states can be modified by the operational environment in order to improve the channel security (cf. AGD_PRE.1 refinement item 2 in (cf. the requirements on Preparative Procedures in 3.5.1.2), the evaluator shall confirm, by following the procedure described in the operational guidance, that this modification can be successfully carried out.
Pass/Fail with Explanation	Pass. This testing was performed in conjunction with FCO_CPC_EXT.1 Test #1.1 to demonstrate correct operation. PSK is configured in the Sensor when a sensor is initialized. This meets the testing requirements.

7.1.4.11 FPT_ITT Test #1

Item	Data
Test Assurance Activity	Test 1: The evaluator shall ensure that communications using each protocol between each pair of authorized TOE components is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FCO_CPC_EXT.1 Test #1.1. External connections from the TOE to other authorized TOE components are sent via an encrypted channel.

7.1.4.12 FPT_ITT Test #2

Item	Data
Test Assurance Activity	Test 2: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FAU_STG_EXT.5 Test #1 and FTP_ITC.1 Test #4. The sensor communicates with the authorized IT entities via an encrypted channel.

7.1.4.13 FPT_ITT Test #3

Item	Data
Test Assurance Activity	Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route between distributed components. The evaluator shall ensure that, for each different pair of non-equivalent component types, the connection is physically interrupted for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) a duration that is shorter than the application layer timeout but is of sufficient length to interrupt the network link layer. The evaluator shall ensure that when physical connectivity is restored, either communications are appropriately protected, or the secure channel is terminated and the registration process (as described in the FTP_TRP.1/Join) re-initiated, with the TOE generating adequate warnings to alert the Security Administrator. In the case that 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 components.

	The interruption shall not be performed at the virtual node (e.g. virtual switch) and must be physical in nature.
Test Steps	• Verify that the sensor and the manager are connected. • Disconnect the cable between the sensor and Ethernet switch for less than the application layer timeout of 60 seconds and verify via PCAPs that the TLS session was not dropped, and that all data continues to be encrypted. The existing TLS connection was maintained, and all data sent between the systems is encrypted. • Disconnect the cable between the sensor and Ethernet switch for more than the application layer timeout of 60 seconds and verify via PCAPs that the TLS session was re-established, and that all data is encrypted. The existing TLS connection was dropped, and a new connection was established before any user data was transmitted.
Expected Test Results	• **TLS connection should be encrypted once the physical connection restores within the application layer timeout and if it exceeded the application layer time.** • **Verify in packet capture.**
Pass/Fail with Explanation	Pass. The TLS connections were maintained or re-established as necessary, and all data sent between the system was encrypted. This satisfies the testing requirements.

7.1.5 IPS Audit

7.1.5.1 FAU_GEN.1/IPS Test#1

Item	Data
Test Assurance Act	The evaluator shall test that the interfaces used to configure the IPS polices yield expected IPS data in association with the IPS policies. A number of IPS policy combination and ordering scenarios need to be configured and tested by attempting to pass both allowed and anomalous network traffic matching configured IPS policies in order to trigger all required IPS events. Note the following: • This activity should have been addressed with a combination of the Test EAs for the other IPS requirements. • As part of testing this activity, the evaluator shall also ensure that the audit data generated to address this SFR can be handled in the manner that FAU_STG_EXT.1 requires for all audit data.
Test Steps	Covered by auditable records in each test case.
Expected Test Results	• The TOE accurately generates audit records for all the required auditable events. • Evidence - Snapshots showing generated logs for audit records.
Pass/Fail with Explanation	Pass. All the auditable events have been recorded.

7.1.6 IPS Policies

7.1.6.1 FMT_SMF.1/IPS Test #1

Item	Data
Test Assurance Act	Test 1: The evaluator shall use the operational guidance to create a signature and enable it on an interface. The evaluator shall then generate traffic that would be successfully triggered by the signature. The evaluator should observe the TOE applying the corresponding reaction in the signature.

Test Steps	• Create a Snort rule for a specific type of traffic. • Enable blocking action for the rule. • Deploy the changes to the sensor. • Simulate traffic matching the rule. • Verify via attack log that the packet is dropped. • Verify via packet capture that the packet is dropped.
Expected Test Results	• **Packet Capture shows that traffic matching the configured signature is dropped by TOE according to applied policy.** • **TOE detects and logs traffic matching the configured signature and drops the traffic according to applied policy.**
Pass/Fail with Explanation	Pass. IPS policy can be created and enabled on an interface. Traffic triggering this policy is detected and the configured reaction is applied. This satisfies testing requirements.

7.1.6.2 FMT_SMF.1/IPS Test #2

Item	Data
Test Assurance Act	Test 2: The evaluator shall then disable the signature and attempt to regenerate the same traffic and ensure that the TOE allows the traffic to pass with no reaction.
Test Steps	• Disable the signature from FMT_SMF.1/IPS Test #1. • Generate the same traffic. • Verify through packet capture and logs that the traffic was appropriately allowed to flow.
Expected Test Results	• **TOE permits traffic matching the configured signature when the applied IDP policy is disabled.** • **Packet Capture shows traffic matching the configured signature being permitted through TOE when the applied IDP policy is disabled.**
Pass/Fail with Explanation	Pass. After disabling the signature, the TOE allows the same traffic to pass through it with no reaction. This meets the testing requirements.

7.1.6.3 FMT_SMF.1/IPS Test #3

Item	Data
Test Assurance Act	Test 3: The evaluator shall use the operational guidance to import signatures and repeat the test conducted in Test 1.
Test Steps	• Upload the attack signature to the Manager device. • Enable the blocking action for the signature. • Apply the signature to the appropriate interface and deploy the changes to the sensor. • Simulate traffic to match the signature. • Verify via logs that the traffic gets blocked. • Verify via packet capture.
Expected Test Results	• **TOE detects and logs traffic matching the configured imported signature and drops the traffic according to applied policy.** • **Packet Capture shows that traffic matching the configured imported signature is dropped by TOE according to applied policy.**

Pass/Fail with Explanation	Pass. The TOE imported signatures, which once enabled, detect traffic matching the signature on the configured interface. This meets the testing requirements.

7.1.6.4 IPS_ABD_EXT.1 Test #1

Item	Data
Test Assurance Act	Test 1: The evaluator shall use the instructions in the operational guidance to configure baselines or anomaly-based rules for each attributes specified in IPS_ABD_EXT.1.1. The evaluator shall send traffic that does not match the baseline or matches the anomaly-based rule and verify the TOE applies the configured reaction. This shall be performed for each attribute in IPS_ABD_EXT.1.1.
Test Steps	• Create a policy that includes examination of many MSSQL Login failures on a single session (IP Source Address, Destination Address and Protocol; TCP Source Port and Destination Port). • Simulate the attack traffic. • Verify with packet capture that the anomalous traffic gets blocked. • Verify via attack log that the anomalous traffic gets blocked. • Create and test a Policy that includes examination of repeated IP fragments with the same offset (IPv4 Flags and Fragment Offset) between the same systems (IP Source Address and Destination Address). • Enable the blocking action for the signature and deploy the changes to the sensor. • Verify with packet capture that the anomalous traffic gets blocked. • Verify via attack log that the anomalous traffic gets blocked. • Verify via attack log. • Create and test a Policy that includes examination of several ICMP Timestamp Requests (IP Protocol, ICMP Type) between the same systems (IP Source Address and Destination Address). • Enable the blocking action for the signature and deploy the changes to the sensor. • Simulate the attack traffic. • Verify with packet capture that the anomalous traffic gets blocked. • Verify via attack log that the anomalous traffic gets blocked. • Create a Policy that includes examination of UDP packets with varying port combinations (IP Protocol, UDP Source Port and Destination Port) between the same systems (IP Source Address and Destination Address) resulting in ICMP Port Unreachable messages (ICMP Type and Code). • Enable the blocking action for the signature and deploy the changes to the sensor. • Simulate the attack traffic. • Verify with packet capture that the anomalous traffic gets responded with ICMP unreachable messages.

	• Verify via attack log that the anomalous traffic gets detected. • Create a Policy that includes examination of TCP Flags on a varying set of connections (IP Source Address, Destination Address and Protocol; TCP Source Port and Destination Port). • Enable the blocking action for the signature and deploy the changes to the sensor. • Simulate the attack traffic. • Verify with packet capture that the anomalous traffic gets blocked. • Verify via attack log that the anomalous traffic gets blocked. • Create a Policy that includes examination of IMAP Login failures due to bruteforcing by protocol header examination on a single connection (IP Source Address, Destination Address and Protocol; TCP Source Port and Destination Port). • Enable the blocking action for the signature and deploy the changes to the sensor. • Simulate the attack traffic. • Verify with packet capture that the anomalous traffic gets blocked. • Verify via attack log that the anomalous traffic gets blocked. • Create a Policy that includes examination of NetBIOS-SS SMB Login failures by protocol header examination on multiple connections (IP Source Address, Destination Address and Protocol; TCP Source Port and Destination Port). • Enable the blocking action for the signature and deploy the changes to the sensor. • Simulate the attack traffic. • Verify with packet capture that the anomalous traffic gets blocked. • Verify via attack log that the anomalous traffic gets blocked. •
Expected Test Results	• **TOE detects and logs traffic matching the baselines or anomaly-based rules for particular attribute.** • **Packet Capture shows that traffic matching the baselines or anomaly-based rules for particular attribute are treated according to applied policy.**
Pass/Fail with Explanation	Pass. The TOE configured baselines, which once enabled, detect traffic matching the baseline on the configured interface. This meets testing requirements.

7.1.6.5 IPS_ABD_EXT.1 Test #2

Item	Data
Test Assurance Act	Test 2: The evaluator shall repeat the test above to ensure that baselines or anomaly- based rules can be defined for each distinct network interface type supported by the TOE.

Pass/Fail with Explanation	Pass. All distinct network interface types supported by the TOE for attributes Throughput, Time of day, Frequency and threshold have been tested as part of IPS_ABD_EXT.1 Test #1

7.1.6.6 IPS_IPB_EXT.1 Test #1

Item	Data
Test Assurance Act	Test 1: The evaluator shall use the instructions in the operational guidance to create a known-bad address list. Using a single IP address, a list of addresses or a range of addresses from that list, the evaluator shall attempt to send traffic through the TOE that would otherwise be allowed by the TOE and observe the TOE automatically drops that traffic.
Test Steps	• Create a firewall policy on the Manager defining a known-bad address along with a known-bad address range. • Enable the blocking action. • Assign the policy to the interface and deploy the changes to the sensor. • Send the traffic from the IPv4 endpoint which was added in the firewall rule. • Verify that the traffic is blocked via logs. • Verify via packet capture. • Send the traffic from the IPv4 range which was added in the firewall rule. • Verify that the traffic is blocked via logs. • Verify via packet capture.
Expected Test Results	• **TOE logs show that traffic matching the single IP address, a list of addresses or a range of addresses in the known-bad address list are dropped by TOE.** • **Packet Capture shows traffic matching the single IP address, a list of addresses or a range of addresses in the known-bad address list are dropped by TOE.**
Pass/Fail with Explanation	Pass. TOE drops traffic matching the configured know bad address list, which would otherwise be allowed. This meets the testing requirements.

7.1.6.7 IPS_IPB_EXT.1 Test #2

Item	Data
Test Assurance Act	Test 2: The evaluator shall use the instructions in the operational guidance to create a known-good address list. Using a single IP address, a list of addresses or a range of addresses from that list, the evaluator shall attempt to send traffic that would otherwise be denied by the TOE and observe the TOE automatically allowing traffic.
Test Steps	• Create a single entry of a known-good address and an additional entry with a range of known-good addresses. • Set the response/action as 'Scan' to allow traffic from the known good IP addresses. • Apply the entries to the TOE's interface and deploy the changes to the sensor. • Send traffic that matches the configured entry of known-good host. • Verify through the TOE's logs that traffic was appropriately permitted. • Verify via packet capture. • Send traffic that matches the configured entry of known-good range. • Verify through the TOE's logs that traffic was appropriately permitted. • Verify via packet capture.

Expected Test Results	• **TOE logs show that after implementation of known-good address list onto a policy, traffic matching the single IP address, a list of addresses or a range of addresses in the address list are permitted by TOE.** • **Packet Capture shows that after implementation of known-good address list onto a policy, traffic matching the single IP address, a list of addresses or a range of addresses in the address list are permitted by TOE.**
Pass/Fail with Explanation	Pass. TOE permits traffic matching the configured know good address and address range. This meets the testing requirements.

7.1.6.8 IPS_IPB_EXT.1 Test #3

Item	Data
Test Assurance Act	Test 3: The evaluator shall add conflicting IP addresses to each list and ensure that the TOE handles conflicting traffic in a manner consistent with the precedence in IPS_NTA_EXT.1.1.
Test Steps	• Add a new known-bad rule below the known-good rule created in IPS_IPB_EXT.1 Test #2 • Configure the new rule to drop traffic from the same IP and IP range as in the known-good rule. • Deploy the changes to the sensor. • Send traffic matching the rules. • Verify through a packet Capture and logs that the traffic is accepted. • Move the new known-bad rule above the known-good rule in the list. • Deploy the changes to the sensor. • Send traffic matching the security policies applied. • Verify through a packet Capture and logs that the traffic is dropped.
Expected Test Results	• **TOE logs show that it handles conflicting traffic according to the order in which rules are applied.** • **Packet Capture shows that it handles conflicting traffic according to the order in which rules are applied.**
Pass/Fail with Explanation	Pass. TOE handles conflicting rules in an Administrator-defined order. This meets the testing requirements.

7.1.6.9 IPS_SBD_EXT.1.1 Test #1

Item	Data
Test Assurance Act	Test 1: The evaluator shall use the instructions in the operational guidance to test that packet header signatures can be created and/or configured with the selected and/or configured reactions specified in IPS_SBD_EXT.1.5 for each of the attributes listed below. Each attribute shall be individually assigned to its own unique signature: • IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; IP Options: and if selected, type of service(ToS)

	• IPv6: Version; payload length; next header; hop limit; source address; destination address; routing header; and, if selected, traffic class and/or flow label. • ICMP: Type; Code; Header Checksum; and Rest of Header (varies based on the ICMP type and code). • ICMPv6: Type; Code; and Header Checksum; and, if selected, Rest of other Header fields (varies based on the ICMP type and code). • TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options. • UDP: Source port; destination port; length; and UDP checksum. The evaluator shall generate traffic to trigger a signature and shall then use a packet sniffer to capture traffic that ensures the reactions of each rule are performed as expected. **TD0722 has been applied**
Test Steps	For each of the attributes: • IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; and IP Options. • IPv6: Version; payload length; next header; hop limit; source address; destination address; and routing header. • ICMP: Type; Code; Header Checksum; ID; Sequence number and other field in the ICMP header (varies based on the ICMP type and code). • ICMPv6: Type; Code; and Header Checksum. • TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options. • UDP: Source port; destination port; length; and UDP checksum. • Configure a filter on the TOE to match the attribute. • Enable the blocking action for the filter and deploy the changes to the sensor. • Simulate traffic to match the configured filter on the TOE. • Verify through a packet capture and through logs that the traffic was appropriately dropped.
Expected Test Results	• **TOE logs traffic matching configured packet header signatures and verifies configured reaction of 'drop' is implemented by TOE.** • **Packet capture verifies the traffic matching configured packet header signatures are dropped.**
Pass/Fail with Explanation	Pass. TOE is triggered with traffic matching configured signatures and reacts in the expected way by dropping the traffic. This meets the testing requirements.

7.1.6.10 IPS_SBD_EXT.1.1 Test #2

Item	Data
Test Assurance Act	The evaluator shall repeat the test above to ensure that signature-based IPS policies can be defined for each distinct network interface type capable of applying signatures as supported by the TOE.

Pass/Fail with Explanation	Pass. Addressed by IPS_SBD_EXT.1.1 Test #1 since only one network interface type is supported for applying signatures.

7.1.6.11 IPS_SBD_EXT.1.2 Test #1

Item	Data
Test Assurance Act	Test 1: The evaluator shall use the instructions in the operational guidance to test that packet payload string-based detection rules can be assigned to the reactions specified in IPS_SBD_EXT.1.5 using the attributes specified in IPS_SBD_EXT.1.2. However, it is not required (nor is it feasible) to test all possible strings of protocol data, the evaluator shall ensure that a selection of strings in the requirement is selected to be tested. At a minimum at least one string using each of the following attributes from IPS_SBD_EXT.1.2 should be tested for each protocol. The evaluator shall generate packets that match the string in the rule and observe the corresponding reaction is as configured. • Test at least one string of characters for ICMPv4 data: beyond the first 4 bytes of the ICMP header. • Test at least one string of characters for ICMPv6 data: beyond the first 4 bytes of the ICMP header. • TCP data (characters beyond the 20 byte TCP header): i. Test at least one FTP (file transfer) command: help, noop, stat,syst, user, abort, acct, allo, appe, cdup, cwd, dele, list, mkd, mode, nlst, pass, pasv, port, pass, quit, rein, rest, retr, rmd, rnfr, rnto, site, smnt, stor, stou, stru, and type. ii. HTTP (web) commands and content: 1) Test both GET and POST commands 2) Test at least one administrator-defined strings to match URLs/URIs, and web page content. iii. Test at least one SMTP (email) state: start state, SMTP commands state, mail header state, mail body state, abort state. *iv.* Test at least one string in any additional attribute type defined within the “other types of TCP payload inspection” assignment, if any other types are specified. • Test at least one string of UDP data: characters beyond the first 8 bytes of the UDP header; • Test at least one string for each additional attribute type defined in the “other types of packet payload inspection” assignment, if any other types are specified.
Test Steps	• Configure the TOE to search for the string SECURITY in an ICMPv4 packet. • Enable the blocking action for the filter and deploy the changes to the sensor. • Send modified traffic that matches the configuration. • Verify through a packet capture and logs that the modified traffic was not allowed through the TOE. • Configure the TOE to search for the string SECURITY in an ICMPv6 packet. • Enable the blocking action for the filter and deploy the changes to the sensor. • Send modified traffic that matches the configuration.

	• Verify through a packet capture and logs that the modified traffic was not allowed through the TOE. • Configure a filter on the TOE to block an FTP user login attempt. • Enable the blocking action for the filter and deploy the changes to the sensor. • Send modified traffic that matches the configuration. • Verify through a packet capture and through logs that a connection was unsuccessful. • Configure the TOE to block HTTP GET packets. • Enable the blocking action for the filter and deploy the changes to the sensor. • Send modified traffic that matches the configuration. • Verify through a packet capture and logs that the modified traffic was not allowed through the TOE. • Configure the TOE to block HTTP POST packets. • Enable the blocking action for the filter and deploy the changes to the sensor. • Send modified traffic that matches the configuration. • Verify through a packet capture and logs that the modified traffic was not allowed through the TOE. • Configure the TOE to block specific string in URLs. • Enable the blocking action for the filter and deploy the changes to the sensor. • Send modified traffic that matches the configuration. • Verify through a packet capture and logs that the modified traffic was not allowed through the TOE. • Configure the TOE to block webpage content. • Enable the blocking action for the filter and deploy the changes to the sensor. • Create a request containing 'portindexl' in its body and simulate. • Verify through a packet capture and logs that attempting to download the zip file was not permitted by the TOE. • Configure the TOE to block any SMTP AUTH packets. • Enable the blocking action for the filter and deploy the changes to the sensor. • Send modified traffic that matches the configuration. • Verify through a packet capture and logs that the modified traffic was not allowed through the TOE. • Configure the TOE to search for the matching data in an UDP packet. • Enable the blocking action for the filter and deploy the changes to the sensor. • Send modified traffic that matches the configuration.

	• Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.
Expected Test Results	• **TOE logs traffic matching payload string-based rules applied and verifies configured reaction of 'drop' is implemented by TOE.** • **Packet capture verifies the traffic matching payload string-based rules applied are dropped.**
Pass/Fail with Explanation	Pass. The TOE logs traffic matching the payload string and reacts to the configured string-based and performs the selected action of dropping the packets.

7.1.6.12 IPS_SBD_EXT.1.2 Test #2

Item	Data
Test Assurance Act	The evaluator shall repeat Test 1 above to ensure that signature-based IPS policies can be defined for each distinct network interface type capable of applying signatures as supported by the TOE.
Pass/Fail with Explanation	Pass. All distinct network interface types capable of applying signatures as supported by the TOE have been tested as part of IPS_SBD_EXT.1.2 Test #1.

7.1.6.13 IPS_SBD_EXT.1.3 Test #1

Item	Data
Test Assurance Act	The evaluator shall create and/or configure rules for each attack signature in IPS_SBD_EXT.1.3. For each attack, the TOE should apply its corresponding signature and enable it to each distinct network interface type capable of applying the signatures. The evaluator shall use packet captures to ensure that the attack traffic is detected by the TOE and a reaction specified in IPS_SBD_EXT.1.5 is triggered and stops the attack. Each attack should be performed one after another so as to ensure that its corresponding signature successfully identified and appropriately reacted to a particular attack.

Test Steps	IP Attacks
	• IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack)
	  ○ Configure the TOE with a rule to detect when the IP fragments overlap.
	  ○ Enable the blocking action for the filter and deploy the changes to the sensor.
	  ○ Send modified traffic that matches the configuration.
	  ○ Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.
	• IP source address equal to the IP destination (Land attack)
	  ○ Configure the TOE with a rule to detect when the IP source address and destination address are equal.
	  ○ Enable the blocking action for the filter and deploy the changes to the sensor.
	  ○ Send modified traffic that matches the configuration.
	  ○ Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.

ICMP Attacks

- Fragmented ICMP Traffic (e.g. Nuke attack)
  - Configure the TOE with a rule to detect ICMP fragmented packets.
  - Enable the blocking action for the filter and deploy the changes to the sensor.
  - Send modified traffic that matches the configuration.
  - Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.

- Large ICMP Packet (e.g. Ping of Death)
  - Configure the TOE with a rule to detect large ICMP Packets.
  - Enable the blocking action for the filter and deploy the changes to the sensor.
  - Send modified traffic that matches the configuration.
  - Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.

TCP Attacks

- TCP NULL Flag
  - Configure the TOE with a rule to detect TCP Null flags.
  - Enable the blocking action for the filter and deploy the changes to the sensor.
  - Send modified traffic that matches the configuration.
  - Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.

- TCP FIN+SYN Flag
  - Configure the TOE a rule to detect TCP FIN+SYN flags.
  - Enable the blocking action for the filter and deploy the changes to the sensor.
  - Send modified traffic that matches the configuration.
  - Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.

- TCP FIN only Flags
  - Configure the TOE a rule to detect TCP FIN flags.
  - Enable the blocking action for the filter and deploy the changes to the sensor.
  - Send modified traffic that matches the configuration.
  - Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.

- TCP SYN+RST Flag
  - Configure the TOE a rule to detect TCP SYN+RST flags.
  - Enable the blocking action for the filter and deploy the changes to the sensor.
  - Send modified traffic that matches the configuration.
  - Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.

UDP Attacks

	• UDP Bomb Attack ○ Configure the TOE a rule to detect UDP bomb attack. ○ Enable the blocking action for the filter and deploy the changes to the sensor. ○ Send modified traffic that matches the configuration. • UDP Chargen DoS Attack ○ Configure the TOE a rule to detect UDP Chargen DoS flags. ○ Enable the blocking action for the filter and deploy the changes to the sensor. ○ Send modified traffic that matches the configuration. ○ Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.
Expected Test Results	• **TOE logs traffic matching payload string-based rules applied and verifies configured reaction of 'drop' is implemented by TOE.** • **Packet capture verifies the traffic matching payload string-based rules applied are dropped.**
Pass/Fail with Explanation	Pass. The TOE configured with payload string-based rules, which once enabled, detect traffic matching the configured payload string and drop the traffic. This meets testing requirements.

7.1.6.14 IPS_SBD_EXT.1.4 Test #1

Item	Data
Test Assurance Act	The evaluator shall configure individual signatures for each attack in IPS_SBD_EXT.1.4. For each attack, the TOE should apply its corresponding signature and enable it to each distinct network interface type capable of applying signatures. The evaluator shall use packet captures to ensure that the attack traffic is detected by the TOE and a reaction specified in IPS_SBD_EXT.1.5 is triggered and stops the attack. Each attack should be performed one after another so as to ensure that its corresponding signature successfully identified and appropriately reacted to a particular attack.
Test Steps	Flooding a host (DoS Attack) • ICMP flooding (Smurf attack, and ping flood) ○ Configure the TOE with a rule to detect when the ICMP Flood attack. ○ Enable the blocking action for the filter and deploy the changes to the sensor. ○ Send modified traffic that matches the configuration. ○ Verify through a packet capture and logs that the modified traffic was not allowed through the TOE. • TCP flooding (e.g. SYN Flood) ○ Configure the TOE with a rule to detect TCP SYN Flood attack. ○ Enable the blocking action for the filter and deploy the changes to the sensor. ○ Send modified traffic that matches the configuration. ○ Verify through a packet capture and logs that the modified traffic was not allowed through the TOE. Flooding a network (DoS Attack)

	• Flooding a network (DoS Attack)
	  ○ Configure the TOE with a rule to detect Network Flood Attack.
	  ○ Enable the blocking action for the filter and deploy the changes to the sensor.
	  ○ Send modified traffic that matches the configuration.
	  ○ Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.
	
	Protocol and Port Scanning
	• IP Protocol Scanning
	  ○ Configure the TOE with a rule to detect IP Protocol Scanning.
	  ○ Enable the blocking action for the filter and deploy the changes to the sensor.
	  ○ Send modified traffic that matches the configuration.
	  ○ Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.
	
	• TCP Port Scanning
	  ○ Configure the TOE with a rule for TCP Port Scanning.
	  ○ Enable the blocking action for the filter and deploy the changes to the sensor.
	  ○ Send modified traffic that matches the configuration.
	  ○ Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.
	
	• UDP Port Scanning
	  ○ Configure the TOE with a rule for UDP Port Scanning.
	  ○ Enable the blocking action for the filter and deploy the changes to the sensor.
	  ○ Send modified traffic that matches the configuration.
	  ○ Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.
	
	• ICMP Scanning
	  ○ Configure the TOE with a rule for ICMP Scanning.
	  ○ Enable the blocking action for the filter and deploy the changes to the sensor.
	  ○ Send modified traffic that matches the configuration.
	  ○ Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.

Expected Test Results	• **TOE logs traffic matching payload string-based rules applied and verifies configured reaction of 'drop' is implemented by TOE.** • **Packet capture verifies the traffic matching payload string-based rules applied are dropped.**
Pass/Fail with Explanation	Pass. TOE handles conflicting rules in an Administrator-defined order. This meets the testing requirements.

7.1.6.15 IPS_SBD_EXT.1.6 Test #1

Item	Data

Test Assurance Act	The evaluator shall repeat one of the tests in IPS_SBD_EXT.1.2 Test 1 but generate multiple non-fragmented packets that contain the string in the rule defined. The evaluator shall verify that the malicious traffic is still detected when split across multiple non-fragmented packets.
Test Steps	• Configure a filter on the TOE to search for the string 'sample.html' split across multiple non-fragmented packets. • Enable the blocking action for the filter and deploy the changes to the sensor. • Send modified traffic that matches the configuration. • Verify through a packet capture and logs that the modified traffic was not allowed through the TOE.
Expected Test Results	• **Packet capture verifies the multiple non-fragmented traffic matching payload string-based rules applied are dropped.** • **TOE logs multiple non-fragmented traffic matching payload string-based rule applied and verifies configured reaction of 'drop' is implemented by TOE.**
Pass/Fail with Explanation	Pass. TOE handles conflicting rules in an Administrator-defined order. This meets the testing requirements.

7.1.7 SSHS

7.1.7.1 FCS_SSHS_EXT.1.2 Test #1

Item	Data
Test Assurance Act	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. **TD0631 has been applied.**
Test Steps	• **Enable public key-based authentication on the TOE.** **SSH-RSA:** • **Generate a public key using SSH-RSA algorithm and note its fingerprint.** • **Copy the generated SSH-RSA public key on the TOE.** • **Log into the TOE with public key-based authentication.** • **Verify the successful connection using TOE logs.** • **Verify the successful connection via packet capture.** **RSA-SHA2-256:** • **Generate a public key using RSA-SHA2-256 algorithm and note its fingerprint.** • **Copy the generated RSA-SHA2-256 public key on the TOE.** • **Log into the TOE with public key-based authentication.**

	• **Verify the successful connection using TOE logs.** • **Verify the successful connection via packet capture.** **RSA-SHA2-512:** • **Generate a public key using RSA-SHA2-512 algorithm and note its fingerprint.** • **Copy the generated RSA-SHA2-512 public key on the TOE.** • **Log into the TOE with public key-based authentication.** • **Verify the successful connection using TOE logs.** • **Verify the successful connection via packet capture.** **ECDSA-SHA2-NISTP256:** • **Generate a public key using ECDSA-SHA2-NISTP256 algorithm and note its fingerprint.** • **Copy the generated ECDSA-SHA2-NISTP256 public key on the TOE.** • **Log into the TOE with public key-based authentication.** • **Verify the successful connection using TOE logs.** • **Verify the successful connection via packet capture.**
Expected Test Results	• **The TOE must successfully establish a SSH session connection with the client using public key authentication.**
Pass/Fail with Explanation	Pass. The TOE is able to establish a SSH session with the client successfully using the supported public key algorithms.

7.1.7.2 FCS_SSHS_EXT.1.2 Test #2

Item	Data
Test Assurance Activity	Test objective: The purpose of these tests is to verify server supports each claimed client authentication method. 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. **TD0631 has been applied.**
Test Steps	• Verify the SSH client's public-key and its fingerprint stored on the TOE. • Configure the SSH client with a new ECDSA keypair for SSH and do not configure the TOE to recognize the client's public key. • Log into the TOE via SSH using ecdsa-256-based authentication. • Verify failed authentication logs on TOE. • Verify authentication failure via packet capture.
Expected Test Results	• **The TOE should reject SSH connections when incorrect/unknown public keys are presented.** • **When a connection is attempted using an ecdsa-sha2-nistp256 public key value that has not been loaded onto the TOE, public key authentication fails, and the user is prompted for a password.**

Pass/Fail with Explanation	Pass. The TOE does not allow a connection when an incorrect public key is presented. This meets the testing requirement.

7.1.7.3 FCS_SSHS_EXT.1.2 Test #3

Item	Data
Test Assurance Activity	Test objective: The purpose of these tests is to verify server supports each claimed client authentication method. 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. **TD0631 has been applied.**
Test Steps	• Enable password-based authentication on the TOE. • Create a user on the TOE and set a password. • Log into the TOE via SSH with password authentication using correct credentials. • Verify using authentication logs on TOE. • Verify via packet capture that SSH session was established.
Expected Test Results	• **The TOE should set up a user with password-based authentication.** • **User authentication succeeds when the correct password is provided by the user.**
Pass/Fail with Explanation	Pass. The TOE accepts password-based authentication from a remote SSH client when the correct password is provided.

7.1.7.4 FCS_SSHS_EXT.1.2 Test #4

Item	Data
Test Assurance Activity	Test objective: The purpose of these tests is to verify server supports each claimed client authentication method. 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. **TD0631 has been applied.**
Test Steps	• Enable password-based authentication on the TOE. • Create a user on the TOE and set a password. • Attempt to Log into the TOE via SSH with correct username and incorrect password and observe the connection getting rejected by the TOE. • Verify authentication via logs that reflect failures. • Verify authentication via packet capture that reflects failures.
Expected Test Results	• **The TOE should set up a user with password-based authentication.** • **User authentication should fail when incorrect password is provided by the user.**
Pass/Fail with Explanation	Pass. The TOE does not allow a connection when an incorrect password is entered.

7.1.7.5 FCS_SSHS_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.
Test Steps	• **Establish an SSH connection to the TOE via acumen-sshs tool and send a packet larger than the established limit.** • **Verify failure via logs.** • **Verify that the large packet is dropped via packet capture.**
Expected Test Results	• **The TOE drops a packet larger than the allowed range.** • **Packet capture shows TOE closes the connection when packet sent is larger than allowed range.**
Pass/Fail with Explanation	Pass. The TOE drops large packets that are received within an SSH session. This meets the testing requirements.

7.1.7.6 FCS_SSHS_EXT.1.4 Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• Initiate an SSH connection from the SSH client to the TOE without enforcing any specific cipher. • Verify that the SSH session was established successfully via log. • Verify that the SSH session was encrypted using only the claimed cipher(s) via packet capture.
Expected Test Results	• **The TOE should establish a SSH session only with the claimed encryption algorithms.** • **The connection must fail in case of unsupported encryption algorithms used.**
Pass/Fail with Explanation	Pass. The TOE is able to establish a SSH session with the client successfully using only the claimed encryption algorithms. This meets the testing requirements.

7.1.7.7 FCS_SSHS_EXT.1.5 Test #1

Item	Data
Test Assurance Activity	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 algorithms. It is sufficient to observe (on the wire) the successful negotiation of the algorithms to satisfy the intent of the test. **TD0631 has been applied.**
Test Steps	• **Verify that the claimed hostkey algorithms are supported by the TOE.** **ECDSA-SHA2-NISTP256:** • Generate an ecdsa-sha2-nistp256 host key pair on the TOE and note its fingerprint. • Login to the TOE using the host public key and verify that the session is established. • Verify via packet capture that the configured host key algorithm was used.
Expected Test Results	• **TOE establishes a successful SSH connection only with the claimed host key algorithms.**
Pass/Fail with Explanation	Pass. The TOE establishes a successful SSH connection using each one of the claimed host public key algorithms.

7.1.7.8 FCS_SSHS_EXT.1.5 Test #2

Item	Data
Test Assurance Activity	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. **TD0631 has been applied.**
Test Steps	• Configure the SSH client to only allow a Host-Key algorithm that is not supported by the TOE. • Attempt to establish an SSH session using the non-claimed host public key algorithm and verify that it fails. • Verify the connection is rejected via logs generated on the TOE. • Verify that the connection is refused via packet capture. •
Expected Test Results	• **TOE should reject a connection request from unclaimed host public key algorithm.**
Pass/Fail with Explanation	Pass. The TOE refuses a SSH connection when a non-supported host public key algorithm is used. This meets the test requirements.

7.1.7.9 FCS_SSHS_EXT.1.6 Test #1

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	NA. The connection uses an implicit MAC. Therefore, this test is not applicable.

7.1.7.10 FCS_SSHS_EXT.1.6 Test #2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	NA. The connection uses an implicit MAC. Therefore, this test is not applicable.

7.1.7.11 FCS_SSHS_EXT.1.7 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure an SSH client to only allow the diffie-hellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.
Test Steps	• Configure the SSH client to only allow the diffie-hellman-group1-sha1 key exchange, that is not supported by the TOE. • Attempt to establish a connection with the TOE from an SSH client using Diffie-hellman-group1-sha1 as the key exchange method and verify that it fails. • Verify connection failure via packet capture. • Verify that the session was not established via logs generated on the TOE. •
Expected Test Results	• The TOE does not permit connections when using diffiehellman-group1-sha1. User gets a 'no matching key exchange method found' error. • Packet capture shows TOE closing connection when kex_algorithm from SSH Client is unsupported. •
Pass/Fail with Explanation	Pass. The SSH connection fails when Diffie-hellman-group1-sha1 (a non-approved algorithm) for the key exchange is used. This meets the testing requirement.

7.1.7.12 FCS_SSHS_EXT.1.7 Test #2

Item	Data
Test Assurance Activity	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.
Test Steps	• Configure the SSH client to only allow key exchange using ecdh-sha2-nistp256 algorithm. • Attempt to establish a connection with the TOE from an SSH client using ecdh-sha2-nistp256 as the key exchange method and verify that it succeeds.

	• Verify the successful connection using packet capture.
Expected Test Results	• **The TOE should make a SSH connection using ecdh-sha2-nistp256 key exchange algorithm.** • **Encrypted packet should flow between the two devices and the packet capture verifies successful SSH connection when using a supported key exchange algorithm.** • **Audit logs should show a successful connection.**
Pass/Fail with Explanation	Pass. The TOE is able to make SSH connections with the claimed key exchange method. This meets the testing requirements.

7.1.7.13 FCS_SSHS_EXT.1.8 Test #1a

Item	Data
Test Assurance Activity	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. 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).
Test Steps	• Set the time-based rekeying threshold for SSH connections to 180 seconds (3 minutes) on the TOE. • Initiate a SSH connection from the SSH client in verbose mode to the TOE and periodically issue the 'show' command till the set threshold has been reached. • Verify the rekeying after the specified time interval. • Verify the time-based rekey logs generated on the TOE.
Expected Test Results	• **The TOE issues a rekey after the specified time as configured on the TOE.** • **Audit logs show the session rekey request has been sent after a time-based threshold has been reached.**
Pass/Fail with Explanation	Pass. The TOE correctly issued a rekey after the time-based threshold had passed.

7.1.7.14 FCS_SSHS_EXT.1.8 Test #1b

Item	Data
Test Assurance Activity	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 traffic-based threshold the evaluator shall use the TOE to connect to an SSH client and shall transmit data to and/or receive data from the TOE within the active SSH session until the threshold for data protected by either encryption key is reached. It is acceptable if the rekey occurs

	before the threshold is reached (e.g. because the traffic is counted according to one of the alternatives given in the Application Note for FCS_SSHS_EXT.1.8). The evaluator shall verify that more data has been transmitted within the SSH session than the threshold allows and shall verify that the TOE initiated a rekey (the method of verification shall be reported by the evaluator). Testing does not necessarily have to be performed with the threshold configured at the maximum allowed value of one gigabyte of transferred traffic but the value used for testing shall not exceed one gigabyte. The evaluator needs to ensure that the rekeying has been initiated by the TOE and not by the SSH client that is connected to the TOE. If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, the evaluator needs to verify that the threshold(s) can be configured as described in the guidance documentation and the evaluator needs to test that modification of the thresholds is restricted to Security Administrators (as required by FMT_MOF.1/Functions). In cases where data transfer threshold could not be reached due to hardware limitations it is acceptable to omit testing of this (SSH rekeying based on data transfer threshold) threshold if both the following conditions are met: 1. An argument is present in the TSS section describing this hardware- based limitation and 2. 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.
Test Steps	• Set the SSH volume-based rekey threshold for TOE to 50MB. • Verify the SSH volume-based rekey threshold for the SSH client. • Use the acumen-sshs-new tool to SSH onto the TOE and start sending traffic via the SSH client. • Verify the rekey after specified volume is sent to the TOE by the client. • Verify the volume-based rekey logs generated on the TOE.
Expected Test Results	• **The TOE must issue a rekey after the specified amount of data is transferred as configured on the TOE.** • **Audit logs must show session rekey requests being sent after reaching the set data limit.**
Pass/Fail with Explanation	Pass. The TOE correctly issued a rekey after the data limit had exceeded the set threshold. This meets the testing requirement.

7.1.8 SSHC

7.1.8.1 FCS_SSHC_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	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. **TD0636 has been applied.**
Test Steps	• Enable public key-based authentication on the SSH server. • Generate a key pair on the TOE using ECDSA-SHA2-NISTP256 algorithm and note its fingerprint. • Copy the generated ECDSA-SHA2-NISTP256 public key on the remote server. • Log into the server from the TOE with public key-based authentication. • Verify the successful connection using TOE logs. • Verify the successful connection via packet capture. •
Expected Test Results	• **The TOE must successfully establish a SSH session connection with the remote non-TOE server using public key authentication.**
Pass/Fail with Explanation	Pass. The TOE is able to establish a SSH session connection with the server successfully using the supported public key algorithms.

7.1.8.2 FCS_SSHC_EXT.1.2 Test #2

Item	Data
Test Assurance Activity	Test objective: The purpose of these tests is to check the authentication of the client to the server using each claimed authentication method. 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. **TD0636 has been applied.**
Test Steps	• Verify that the password-based authentication is enabled on the TOE. • Log into the SSH server from the TOE via SSH with password authentication with correct password. • Verify the successful password-based authentication logs. • Verify via packet capture that SSH session was established.
Expected Test Results	• **User authentication should succeed when the correct password is provided by the user.** • **Audit logs must show successful login of user with password.**
Pass/Fail with Explanation	Pass. The TOE is able to establish a successful SSH connection when the correct password is provided for password-based authentication. This meets the testing requirements.

7.1.8.3 FCS_SSHC_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.
Test Steps	• **Run the acumen-sshc tool on the SSH server, using the -s option to send a large packet.** • **Initiate an SSH connection from the TOE and verify large packet being sent to the TOE by the acumen-sshc tool.**

	• **Verify via packet capture that the packet is dropped and the connection terminated.** • **Verify through logs that the connection was terminated.**
Expected Test Results	• **The TOE should drop packets larger than the allowed range.** • **Log showing the reason for closing the connection.** • **Packet capture showing TOE closes the connection when packet sent is larger than allowed range.**
Pass/Fail with Explanation	Pass. The TOE drops large packets that are received within an SSH session.

7.1.8.4 FCS_SSHC_EXT.1.4 Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• Initiate a SSH connection from the TOE to the non-TOE SSH server without enforcing any specific cipher. • Verify that the SSH session was established successfully via log. • Verify that the SSH session was encrypted using only the claimed cipher(s) via packet capture.
Expected Test Results	• **Evidence (screenshot or CLI output) showing configuration of each algorithm.** • **Log show configuration of each algorithm.** • **Packet capture of each session establishment.**
Pass/Fail with Explanation	Pass. The TOE is able to establish a SSH session with a non-TOE server successfully using only the claimed encryption algorithms. This meets the testing requirements.

7.1.8.5 FCS_SSHC_EXT.1.5 Test #1

Item	Data
Test Assurance Activity	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. **TD0636 has been applied.**
Test Steps	• **Verify that the claimed hostkey algorithms are supported by the SSH server.** **SSH-RSA:** • **Generate a ssh-rsa host key pair on the SSH server and note its fingerprint.** • **Initiate a connection from the TOE to the SSH server using the host public key and verify that the session is established.** • **Verify via logs that the SSH session was initiated successfully.** • **Verify via packet capture that the configured host key algorithm was used.** **RSA-SHA2-256:** • **Generate a rsa-sha2-256 host key pair on the SSH server and note its fingerprint.** • **Initiate a connection from the TOE to the SSH server using the host public key and verify that the session is established.** • **Verify via logs that the SSH session was initiated successfully.** • **Verify via packet capture that the configured host key algorithm was used.** **RSA-SHA2-512:** • **Generate a rsa-sha2-512 host key pair on the SSH server and note its fingerprint.** • **Initiate a connection from the TOE to the SSH server using the host public key and verify that the session is established.** • **Verify via logs that the SSH session was initiated successfully.** • **Verify via packet capture that the configured host key algorithm was used.** **ECDSA-SHA2-NISTP256:** • **Generate an ecdsa-sha2-nistp256 host key pair on the SSH server and note its fingerprint.** • **Initiate a connection from the TOE to the SSH server using the host public key and verify that the session is established.** • **Verify via logs that the SSH session was initiated successfully.** • **Verify via packet capture that the configured host key algorithm was used.**
Expected Test Results	• **TOE should establish a successful SSH connection only with the claimed host key algorithms.**
Pass/Fail with Explanation	Pass. The TOE establishes a successful SSH connection using each one of the claimed host public key algorithms with the SSH server.

7.1.8.6 FCS_SSHC_EXT.1.5 Test #2

Item	Data
Test Assurance Activity	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.

	TD0636 has been applied.
Test Steps	• Configure the SSH server to only allow a Host-Key algorithm that is not supported by the TOE. • Initiate a connection from the TOE to the SSH server. • Verify that the connection is refused via packet capture. • Verify that the connection is refused via logs. •
Expected Test Results	• **The SSH connection attempt using an unclaimed host public key algorithm must not be initiated from the TOE.**
Pass/Fail with Explanation	Pass. The TOE does not support a SSH session initiation with unclaimed public host-key algorithm.

7.1.8.7 FCS_SSHC_EXT.1.6 Test #1

Item	Data
Test Assurance Activity	[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.
Pass/Fail with Explanation	NA. The connection uses an implicit MAC. Therefore, this test is not applicable.

7.1.8.8 FCS_SSHC_EXT.1.6 Test #2

Item	Data
Test Assurance Activity	[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.
Pass/Fail with Explanation	NA. The connection uses an implicit MAC. Therefore, this test is not applicable.

7.1.8.9 FCS_SSHC_EXT.1.7 Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• **Verify the supported key exchange algorithms on the SSH server.** • **Establish an SSH session from the TOE to the SSH server with the supported key exchange algorithm (ecdh-sha2-nistp256).** • **Verify through logs that the connection was established.**

	• **Verify the successful connection via packet capture.**
Expected Test Results	• **The TOE should make a SSH connection using ecdh-sha2-nistp256 key exchange algorithm.** • **Encrypted packet should flow between the two devices and the packet capture verifies successful SSH connection when using a supported key exchange algorithm.** • **Audit logs should show a successful connection.**
Pass/Fail with Explanation	Pass. The TOE is able to initiate a SSH session with the SSH server with the claimed key exchange method. This meets the testing requirements.

7.1.8.10 FCS_SSHC_EXT.1.8 Test #1a

Item	Data
Test Assurance Activity	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. 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).
Test Steps	• Set the time-based rekeying threshold to 120 seconds (2 minutes) on the TOE. • Initiate a SSH connection from the TOE to the SSH server in verbose mode and periodically issue the 'date' command till the set threshold has been reached. • Verify the rekeying after the specified time interval. • Verify the time-based rekey logs generated on the TOE.
Expected Test Results	• **The TOE should issue a rekey after the specified time as configured on the TOE.** • **Audit logs should reflect the session rekey messages after the time-based threshold has been crossed.**
Pass/Fail with Explanation	Pass. The TOE correctly issues a rekey after the specified time period has been crossed.

7.1.8.11 FCS_SSHC_EXT.1.8 Test #1b

Item	Data
Test Assurance Activity	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 traffic-based threshold the evaluator shall use the TOE to connect to an SSH server and shall transmit data to and/or receive data from the TOE within the active SSH session until the threshold for data protected by either encryption key is reached. It is acceptable if the rekey occurs before the threshold is reached (e.g. because the traffic is counted according to one of the alternatives given in the Application Note for FCS_SSHC_EXT.1.8). 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: 1) An argument is present in the TSS section describing this hardware- based limitation and 2) 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.
Test Steps	• Set the SSH volume-based rekey threshold for TOE to 1GB. • Initiate a new SSH session on TOE and fetch a file more than the set threshold from a non-TOE entity using sftp in verbose mode. • Verify via logs that rekey takes place after set the data limit.
Expected Test Results	• **The TOE must issue a rekey after the specified amount of data is transferred as configured on the TOE.** • **Audit logs must show session rekey requests being sent after reaching the set data limit.**
Pass/Fail with Explanation	Pass. The TOE correctly issued a rekey after the data limit had exceeded the set threshold. This meets the testing requirement.

7.1.8.12 FCS_SSHC_EXT.1.9 Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• **Delete the 'known_hosts' file to remove all entries in the TOE's list of recognized SSH server host keys.** • **From the TOE, attempt to establish a SSH connection to an SSH server.** • **Verify that the authenticity for the SSH server fails and the TOE prompts a message for the user to accept the new key before continuing the connection.** • **Verify the connection via logs.** • **Verify the connection via packet capture.**
Expected Test Results	• Evidence (screenshot or CLI output) of configuring each type of key-based authentication. • Packet capture of each session being rejected. •
Pass/Fail with Explanation	Pass. The TOE prompts the user to accept or deny the SSH server key before continuing the connection.

7.1.8.13 FCS_SSHC_EXT.1.9 Test #2

Item	Data
Test Assurance Activity	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.
Test Steps	• Load the SSH server hostkey into the TOE's local database. • Change the SSH server hostkey pair without loading it into the TOE. • Attempt SSH connection to the SSH server and verify the connection is refused by the TOE. • Verify the connection failure via packet capture. • Verify the connection failure via logs.
Expected Test Results	• **Evidence (screenshot or CLI output) of configuring each type of key-based authentication.** • **Packet capture of each session being rejected.**
Pass/Fail with Explanation	Pass. The TOE does not allow the SSH connection when the server host key changes.

7.1.9 TLSC

7.1.9.1 FCS_TLSC_EXT.2.1

Item	Data

Test Assurance Activity	The evaluator shall establish a connection to a peer server that is configured for mutual authentication (i.e. sends a server Certificate Request (type 13) message). The evaluator observes that the TOE TLS client sends both client Certificate (type 11) and client Certificate Verify (type 15) messages during its negotiation of a TLS channel and that Application Data is sent. **TD0670 is applied.**
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FIA_X509_EXT.1.1/ ITT Test #1a. The TOE allows a successful connection when valid certificates are presented by the server.

7.1.10 Update

7.1.10.1 FTP_TST_EXT.1 Test #1

Item	Data
Test Assurance Activity	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. 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. 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.
Test Steps	• Login to the TOE through console and issue the command to reboot the TOE. • Verify the expected self-tests are being performed by the TOE.
Expected Test Results	• **The TOE should execute all claimed self-tests during bootup.** • **Evidence (screenshot or CLI output) showing successful self-tests.**
Pass/Fail with Explanation	Pass. The TOE performs all the claimed self-tests successfully. This meets the testing requirements.

7.1.10.2 FPT_TUD_EXT.1 Test #1

Item	Data
Test Assurance Activity	The evaluator performs the version verification activity to determine the current version of the product as well as the most recently installed version (should be the same version before updating). 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.
Test Steps	• Show the current version. • Upgrade the sensor using the 'loadimage' command. • Show the new version post upgrade and reboot. • Verify the successful upgrade via logs.
Expected Test Results	• **The TOE should successfully update the current version with the new version after verifying the integrity of the new image.** • **Evidence - screenshot showing new version post upgrade.**
Pass/Fail with Explanation	Pass. The TOE successfully updates the software image when a legitimate image is used. This meets the testing requirements.

7.1.10.3 FPT_TUD_EXT.1 Test #2(a)

Item	Data
Test Assurance Activity	Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below, and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates: 1) A modified version (e.g. using a hex editor) of a legitimately signed update 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.
Test Steps	• Verify the current version of the TOE. • Using a Hex editor modify an otherwise good firmware image. • Import the corrupt image on the manager. • Attempt to import the corrupt image on the manager and verify that this fails. • Verify via logs that the upgrade fails. • Verify that the TOE version remains unchanged.
Expected Test Results	• **The TOE should detect and reject the modified image for software update.** • **Evidence – TOE logs showing software upgrade failure logs.**
Pass/Fail with Explanation	Pass. The TOE detects and rejects the modified image for software update.

7.1.10.4 FPT_TUD_EXT.1 Test #2(b)

Item	Data
Test Assurance Activity	[conditional]: If **the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE** the following test shall be performed (otherwise the test shall be omitted). The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below, and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates: 2) An image that has not been signed 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.
Test Steps	• Verify the current software version on the TOE. • Try to update the TOE using an unsigned image and observe the failure. • Verify the update failure using logs. • Verify that the TOE version remains unchanged.
Expected Test Results	• **The TOE should detect and reject the image without signature for a software update.** • **Software upgrade failed logs generated on TOE.**
Pass/Fail with Explanation	Pass. The TOE software was able to detect when an image was not signed and rejected the image. This meets the testing requirements.

7.1.10.5 FPT_TUD_EXT.1 Test #2(c)

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

	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) 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.
Test Steps	• Verify the current image version on the TOE. • Using the manager GUI, import a sensor image with an invalid signature. • Attempt to deploy the imported image and verify that it fails. • Verify the logs reflecting failure. • Verify the TOE image was not upgraded.
Expected Test Results	• **The TOE should detect and reject the image with an invalid signature for the software update.** • **Software upgrade failed logs generated on TOE.**
Pass/Fail with Explanation	Pass. The TOE was able to detect when an image with an invalid signature was used for upgrading and rejected the upgrade. This meets the testing requirements.

7.1.10.6 FPT_TUD_EXT.1 Test #2 (d)

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	NA. The TOE does not support delayed activation of updates.

7.1.10.7 FPT_TUD_EXT.1 Test #3 (a)

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

	determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test: 1) The evaluator obtains or produces an illegitimate update such that the hash of the update does not match the published hash. The evaluator provides the published hash value to the TOE and calculates the hash of the update either on the TOE itself (if that functionality is provided by the TOE), or else outside the TOE. The evaluator confirms that the hash values are different, and attempts to install the update on the TOE, verifying that this fails because of the difference in hash values (and that the failure is logged). Depending on the implementation of the TOE, the TOE might not allow the Security Administrator to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE. 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.
Pass/Fail with Explanation	NA. The TOE does not support published hash verification.

7.1.10.8 FPT_TUD_EXT.1 Test #3 (b)

Item	Data
Test Assurance Activity	[conditional]: If **the TOE itself verifies a hash value over an image against a published hash value (i.e. reference value) that has been imported to the TOE from outside such that the TOE itself authorizes the installation of an image to update the TOE,** the following test shall be performed (otherwise the test shall be omitted. If the published hash is provided to the TOE by the Security Administrator and the verification of the hash value over the update file(s) against the published hash is performed by the TOE, then the evaluator shall perform the following tests. The evaluator first confirms that no update is pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test: 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. 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.
Pass/Fail with Explanation	NA. The TOE does not support published hash verification.

7.1.10.9 FPT_TUD_EXT.1 Test #3 (c)

Item	Data
Test Assurance Activity	[conditional]: If **the TOE itself verifies a hash value over an image against a published hash value (i.e. reference value) that has been imported to the TOE from outside such that the TOE itself authorizes the installation of an image to update the TOE,** the following test shall be performed (otherwise the test shall be omitted. If the published hash is provided to the TOE by the Security Administrator and the verification of the hash value over the update file(s) against the published hash is performed by the TOE, then the evaluator shall perform the following tests. The evaluator first confirms that no update is pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test: 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.
Pass/Fail with Explanation	NA. The TOE does not support delayed activation of updates.

7.1.11 X509-ITT

7.1.11.1 FIA_X509_EXT.1.1/ITT Test #1a

Item	Data

Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).
Test Steps	• Generate a CSR on the manager for the sensor. • Sign the CSR and load the certificate chain onto the Manager. • Verify logs indicating successful certificate import onto the Manager. • Configure the Manager to communicate with the Sensor and verify that this is successful. • Verify the successful connection via packet capture.
Expected Test Results	**The TOE allows for a successful connection when a complete chain is present.**
Pass/Fail with Explanation	Pass. The TOE allows for a successful connection when a complete chain is present.

7.1.11.2 FIA_X509_EXT.1.1/ITT Test #1b

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.
Test Steps	• Remove the ICA from the manager's certificate chain used in Test#1a. • Import the invalid manager certificate on the manager GUI. • Verify that the sensor's trust with the manager breaks leading to connection failure. • Verify that the connection was not established using TOE's logs. • Verify the unsuccessful connection via packet capture.

Expected Test Results	**When a complete chain of certificates is not presented, the TOE doesn't establish a successful TLS connection. The packet capture depicts that a FIN packet has been sent to indicate that the connection has been terminated.**
Pass/Fail with Explanation	Pass. When a complete chain of certificates is not presented, the TOE doesn't establish a successful TLS connection.

7.1.11.3 FIA_X509_EXT.1.1/ITT Test #2

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.
Test Steps	• Create an expired manager certificate. • Import the expired manager certificate on the manager GUI. • Verify that the sensor's trust with the manager breaks leading to connection failure. • Verify that the connection was not established using TOE's logs. • Verify the unsuccessful connection via packet capture.
Expected Test Results	**When an expired certificate is presented, the TOE terminates the TLS connection. The packet capture shows that the connection was terminated, and a FIN and RST packet were sent to indicate that connection was not established successfully.**
Pass/Fail with Explanation	Pass. When an expired certificate is presented, the TOE terminates the TLS connection.

7.1.11.4 FIA_X509_EXT.1.1/ITT Test #3

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.

Pass/Fail with Explanation	NA. No revocation checking is specified for this component.

7.1.11.5 FIA_X509_EXT.1.1/ITT Test #4

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for selftesting is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols: Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.
Pass/Fail with Explanation	NA. No revocation checking is specified for this component.

7.1.11.6 FIA_X509_EXT.1.1/ITT Test #5

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for selftesting is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols: Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)
Test Steps	• Observe a successful TLS exchange between the Manager and sensor and note the fixed hex bytes that precede the first bytes of the Manager certificate. • Terminate the trust channel between the sensor and Manager. • Pass the previously noted fixed bytes, offset and new data bytes for the AcumenMITM tool to replace the first bytes of the certificate with. • Initiate a new TLS trust channel between the devices. • Verify that the modification happens and the TLS channel and hence the trust establishment fails. • Verify the failure via packet capture. • Verify the failure via logs.
Expected Test Results	**The TOE denies a TLS connection when it is presented with a certificate that has been modified using the AcumenMITM tool. The tool modifies the first few bytes of the certificate. The packet capture verifies that the connection is not established due to the bad certificate.**
Pass/Fail with Explanation	Pass. The TOE denies a TLS connection when it is presented with a certificate whose first bytes have been modified.

7.1.11.7 FIA_X509_EXT.1.1/ITT Test #6

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for selftesting is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols:

	Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC5280 Sec. 4.1.1.3), which is normally the last field in the certificate, and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
Test Steps	• Observe a successful TLS exchange between the Manager and sensor and note the fixed hex bytes that precede the last bytes of the Manager certificate. • Terminate the trust channel between the sensor and Manager. • Pass the previously noted fixed bytes, negative offset and new data bytes for the AcumenMITM tool to replace the last bytes of the certificate with. • Initiate a new TLS trust channel between the devices. • Verify that the modification happens and the TLS channel and hence the trust establishment fails. • Verify the failure via packet capture. • Verify the failure via logs.
Expected Test Results	**The TOE fails to establish a TLS connection when the last bytes in the signatureValue field of the certificate are modified using the AcumenMITM tool. The packet capture proves that there is a decrypt error and the logs show that there is a failure in establishing connection due to certificate signature failure.**
Pass/Fail with Explanation	Pass. The TOE fails to establish a TLS connection when the last bytes in the signatureValue field of the certificate are modified.

7.1.11.8 FIA_X509_EXT.1.1/ITT Test #7

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for selftesting is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols: Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.)
Test Steps	• Observe a successful TLS exchange between the Manager and sensor, and note the fixed hex bytes that precede the public key bytes of the Manager certificate. • Terminate the trust channel between the sensor and Manager. • Pass the previously noted fixed bytes, offset and new data bytes for the AcumenMITM tool to replace the public key bytes of the certificate with. • Initiate a new TLS trust channel between the devices. • Verify that the modification happens and the TLS channel and hence the trust establishment fails. • Verify the failure via packet capture. • Verify the failure via logs.

Expected Test Results	**The TOE rejects a TLS connection that is forged using the AcumenMITM tool to modify the certificate such that its public key is modified and uses the same certificate for establishing the TLS connection.**
Pass/Fail with Explanation	Pass. The TOE rejects a TLS connection that is forged using the AcumenMITM tool to modify the certificate such that its public key is modified.

7.1.11.9 FIA_X509_EXT.1.1/ITT Test #8a

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** **(Conditional on TOE ability to process CA certificates presented in certificate message)** The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain. **TD0527 (12/1 Update) has been applied.**
Pass/Fail with Explanation	NA. EC signatures are not supported for the TLS functionality .

7.1.11.10 FIA_X509_EXT.1.1/ITT Test #8b

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** **(Conditional on TOE ability to process CA certificates presented in certificate message)** The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid. **TD0527 (12/1 Update) has been applied.**
Pass/Fail with Explanation	NA. EC signatures are not supported for the TLS functionality

7.1.11.11 FIA_X509_EXT.1.1/ITT Test #8c

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)**

	The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store. **TD0527 (12/1 Update) has been applied.**
Pass/Fail with Explanation	NA. EC signatures are not supported for the TLS functionality

7.1.11.12 FIA_X509_EXT.1.2/ITT Test #1

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions n FIA_X509_EXT.2.1/ITT. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted. The goal of the following tests it to verify that the TOE accepts only certificates that have been marked as CA certificates by using basicConstraints with the CA flag set to True (and implicitly that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation). For each of the following tests the evaluator shall create a chain of at least two certificates: a self-signed root CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain). Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).
Test Steps	• Create an ICA Certificate with no Basic Constraints extension. • Replace the good ICA in the sensor certificate chain with the above one. • Attempt to load the modified certificate chain to the TOE and verify that it gets rejected. • Verify that the certificates are rejected using logs.
Expected Test Results	• **The TOE should reject certificates signed by CA that does not contain the BasicConstraints Extension.** • **Failure logs on the TOE showing it discards the certificate.**
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that do not contain the Basic Constraints extension. This meets the testing requirements.

7.1.11.13 FIA_X509_EXT.1.2/ITT Test #2

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions n FIA_X509_EXT.2.1/ITT. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted. The goal of the following tests it to verify that the TOE accepts only certificates that have been marked as CA certificates by using basicConstraints with the CA flag set to True (and implicitly that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation). For each of the following tests the evaluator shall create a chain of at least two certificates: a self-signed root CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain). Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).
Test Steps	• Create an ICA Certificate with the CA flag in the Basic Constraints extension set to FALSE. • Replace the good ICA in the sensor certificate chain with the above one. • Attempt to load the modified certificate chain to the TOE and verify that it gets rejected. • Verify that the certificates are rejected using logs.
Expected Test Results	**The TOE rejects a certificate where the CA certificate contains the CA flag in the Basic Constraints extension set to FALSE. The logs depict the rejection.**
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that has the CA flag in the Basic Constraints extension set to FALSE. This meets the testing requirements.

7.1.11.14 FIA_X509_EXT.2/ITT Test #1

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each trusted channel: The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the guidance

	documentation to determine that all supported administrator-configurable options behave in their documented manner.
Pass/Fail with Explanation	NA. No revocation checking is specified for this component.

7.1.11.15 FIA_X509_EXT.1.3/ITT Test #1

Item	Data
Test Assurance Activity	Test 1: The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated request and ensure that it conforms to the format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.
Test Steps	• Generate a CSR on the TOE. • Examine the CSR contents using XCA and verify that it contains the following subject fields: Common Name, Organization, Organizational Unit, Country. Also verify that it contains details about the public key and other details
Expected Test Results	**The TOE is able to generate a CSR with the required fields.**
Pass/Fail with Explanation	Pass. The TOE is able to generate a CSR with all of the required information. This meets the testing requirements.

7.1.11.16 FIA_X509_EXT.1.3/ITT Test #2

Item	Data
Test Assurance Activity	Test 2: The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the response message, and demonstrate that the function succeeds.
Test Steps	• Generate a CSR on the TOE. • Export and sign the generated CSR using the ICA. • Remove the ICA certificate from the TOE certificate chain. • Attempt to load the above invalid certificate chain on the TOE. • Verify that the TOE rejects the certificate chain because the full trust chain is not present. • Verify that logs confirm the failure in loading of TOE certificates. • Add back the ICA to the TOE certificate chain to ensure that it now has a full certificate path. • Re-attempt to load the now valid certificate chain on the TOE. • Verify that the TOE accepts the certificate because the path validation succeeded. • Verify that logs confirm the successful loading of TOE certificates.
Expected Test Results	**The TOE will reject signed certificates if there is no trust chain or if the chain is broken; the TOE will accept certificates for full valid chains.**
Pass/Fail with Explanation	Pass. The TOE does not install CSR responses signed by a CA without a full trust path. This meets the testing requirements.

7.2 Manager

7.2.1 Audit

7.2.1.1 FAU_GEN.1 Test#1

Item	Data
Test Assurance Activity	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. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.
Test Steps	• Trigger each auditable event on the TOE. Verify that each audit record is generated and contains the required information
Expected Test Results	• **The TOE accurately generates audit records for all the required auditable events** • **Evidence- Snapshot showing generated logs for audit records.**
Pass/Fail with Explanation	Pass. The audit records associated with each test case are recorded with each test case. A comparison of required audit records to the presented audit records was additionally performed. This analysis shows that each required audit record is generated by the TOE.

7.2.1.2 FAU_STG_EXT.1 Test#1

Item	Data
Test Assurance Activity	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.
Test Steps	• Verify the openssl details used on the syslog server. • Login to the manager and configure syslog server. • Establish a connection with the syslog server using openssl. • Perform various auditable events. • Confirm that each event has been logged on the syslog server.

	Verify via packet capture that syslog messages have been sent encrypted.
Expected Test Results	**The TOE sends logs to syslog server via TLS.**
Pass/Fail with Explanation	Pass. The TOE is capable of transferring audit data to an external audit server automatically without administrator intervention. This meets the testing requirements.

7.2.1.3 FAU_STG_EXT.1 Test#2(a)

Item	Data
Test Assurance Activity	Test 2: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.3. Depending on the configuration this means that the evaluator has to check the content of the audit data when the audit data is just filled to the maximum and then verifies that: 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).
Pass/Fail with Explanation	NA. The TOE overwrites previous audit as claimed in the ST.

7.2.1.4 FAU_STG_EXT.1 Test#2(b)

Item	Data
Test Assurance Activity	Test 2: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.3. Depending on the configuration this means that the evaluator has to check the content of the audit data when the audit data is just filled to the maximum and then verifies that: 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)
Test Steps	• Configure the smallest possible logging space on the TOE. • Find the timestamp of the oldest log of the TOE. • Generate lots of audit records until the set threshold is met and now verify the timestamp of the latest log. • Verify that the TOE overwrites the oldest audit records.
Expected Test Results	**The TOE overwrites the oldest log when the log buffer reaches its set limit.**
Pass/Fail with Explanation	Pass. The TOE overwrites the oldest logs when the log buffer reaches its set limit.

7.2.1.5 FAU_STG_EXT.1 Test#2(c)

Item	Data
Test Assurance Activity	The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.3. Depending on the configuration this means that the evaluator has to check the content of the audit data when the audit data is just filled to the maximum and then verifies that: The TOE behaves as specified (for the option '**other action**' in FAU_STG_EXT.1.3).
Pass/Fail with Explanation	NA. The TOE overwrites previous audit as claimed in the ST.

7.2.1.6 FPT_STM_EXT.1 Test#1

Item	Data
Test Assurance Activity	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.
Test Steps	• Confirm the current time. • Set a new time on the manager. • Verify that the time on the TOE was updated. • Verify logs were generated for the time change.
Expected Test Results	• **Logs successfully reflect changed time according to the set time on TOE.** • **Snapshot showing updated time.** • **TOE should generate logs for the time change.**
Pass/Fail with Explanation	Pass. Observed that Security Admin is able to modify time on TOE.

7.2.1.7 FPT_STM_EXT.1 Test#2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	NA. TOE does not support the use of an NTP server.

7.2.1.8 FPT_STM_EXT.1 Test#3

Item	Data
Test Assurance Activity	[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.
Pass/Fail with Explanation	NA. TOE is not a vND.

7.2.1.9 FTP_ITC.1 Test#1

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. This test has been performed in conjunction with FAU_STG_EXT.1 Test #1 and FCS_TLSC_EXT.1 tests. External connections from the TOE are sent via an encrypted channel. This meets the testing requirements.

7.2.1.10 FTP_ITC.1 Test#2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. This test is performed during the course of FTP_ITC.1 Test# 1, FAU_STG_EXT.1 and FCS_TLSC_EXT.1 tests. The TOE connects with all claimed communications channels. All data sent within these communication channels are also not sent in plaintext.

7.2.1.11 FTP_ITC.1 Test#3

Item	Data
Test Assurance Activity	The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
Pass/Fail with Explanation	Pass. This test is performed during the course of FTP_ITC.1 Test# 1, FAU_STG_EXT.1 and FCS_TLSC_EXT.1 tests. The TOE connects with all claimed communications channels. All data sent within these communication channels are also not sent in plaintext.

7.2.1.12 FTP_ITC.1 Test#4

Item	Data
Test Assurance Activity	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: 1. A duration that exceeds the TOE's application layer timeout setting,

	2. 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.
Test Steps	• Set the application layer timeout for the TOE. • Configure a syslog server on the TOE. • Initiate a connection with the syslog server. • Disconnect the cable between the TOE and the syslog server for approximately 1 minute (less than the application layer timeout of 4.5 minutes) and verify via PCAPs that the TLS session was not dropped, and that all data continues to be encrypted. The existing TLS connection was maintained, and all data sent between the systems is encrypted. • Disconnect the cable between the TOE and the syslog server for approximately 5 minutes (more than the application layer timeout of 4.5 minutes) and verify via PCAPs that the TLS session was re-established, and that all data is encrypted. The existing TLS connection was dropped, and a new connection was established before any user data was transmitted.
Expected Test Results	• **The TOE should send encrypted data before and after the temporary disconnection with the external IT entity.**
Pass/Fail with Explanation	Pass. The TLS connections were maintained or re-established as necessary, and all data sent between the system was encrypted. This satisfies the testing requirements.

7.2.2 Crypto

7.2.2.1 FCS_CKM.2 DH14

This test was removed by TD0580.

7.2.2.2 FCS_CKM.1 RSA

Item	Data
Test Assurance Activity	Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). **Key Generation for FIPS PUB 186-4 RSA Schemes** The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent *e*, the private prime factors *p* and *q*, the public modulus *n* and the calculation of the private signature exponent d.

	Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include: c) Random Primes: • Provable primes • Probable primes d) Primes with Conditions: • Primes p1, p2, q1, q2, p and q shall all be provable primes • Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes • Primes p1, p2, q1, q2, p and q shall all be probable primes To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation.
Pass/Fail with Explanation	Algorithm: RSA KeyGen Key size / Modulus: 2048 CAVP #: A4660, A2624 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.2.3 FCS_CKM.1 ECC

Item	Data
Test Assurance Activity	Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). **Key Generation for Elliptic Curve Cryptography (ECC)** FIPS 186-4 ECC Key Generation Test For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation. FIPS 186-4 Public Key Verification (PKV) Test For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Pass/Fail with Explanation	Algorithm: ECDSA KeyGen, ECDSA KeyVer Curves: P-256, P-384 CAVP #: A4660, A2624 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.2.4 FCS_CKM.2 SP800-56A

Item	Data

Test Assurance Activity	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 The evaluator shall verify a TOE's implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag. Function Test The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested. The evaluator shall obtain the DKM, the corresponding TOE's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields. If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret. The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values. If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm. Validity Test The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOE's public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE id fields.

	The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOE's static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass). The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results with the results using a known good implementation verifying that the TOE detects these errors.
Pass/Fail with Explanation	Algorithm: KAS-ECC Sp800-56Ar3 Curves: P-256, P-384 CAVP #: A4660 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.2.5 FCS_COP.1/DataEncryption AES-GCM

Item	Data
Test Assurance Activity	AES-GCM Test The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths: 128 bit and 256 bit keys b) Two plaintext lengths. One of the plaintext lengths shall be a nonzero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported. c) Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported. d) Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested. The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known. The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.
Pass/Fail with Explanation	Algorithm: AES GCM Key size: 128, 256 CAVP #: A4660, A2624 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.2.6 FCS_COP.1/SigGen ECDSA

Item	Data
Test Assurance Activity	**ECDSA Algorithm Tests** **ECDSA FIPS 186-4 Signature Generation Test** For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator sha long messages and obtain for each message a public key and the resulting signature values R and S. T the evaluator shall use the signature verification function of a known good implementation. **ECDSA FIPS 186-4 Signature Verification Test** For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator sha 1024-bit message, public key and signature tuples and modify one of the values (message, public key 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Pass/Fail with Explanation	Algorithm: ECDSA SigGen, ECDSA SigVer Curves: P-256 CAVP #: A4660, A2624 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.2.7 FCS_COP.1/SigGen RSA

Item	Data
Test Assurance Activity	RSA Signature Algorithm Tests Signature Generation Test The evaluator generates or obtains 10 messages for each modulus size/SHA combination supported by the TOE. The TOE generates and returns the corresponding signatures. The evaluator shall verify the correctness of the TOE's signature using a trusted reference implementation of the signature verification algorithm and the associated public keys to verify the signatures. Signature Verification Test For each modulus size/hash algorithm selected, the evaluator generates a modulus and three associated key pairs, (d, e). Each private key d is used to sign six pseudorandom messages each of 1024 bits using a trusted reference implementation of the signature generation algorithm. Some of the public keys, e, messages, or signatures are altered so that signature verification should fail. For both the set of original messages and the set of altered messages: the modulus, hash algorithm, public key e values, messages, and signatures are forwarded to the TOE, which then attempts to verify the signatures and returns the verification results. The evaluator verifies that the TOE confirms correct signatures on the original messages and detects the errors introduced in the altered messages.
Pass/Fail with Explanation	Algorithm: RSA SigGen, RSA SigVer Key size / Modulus: 2048 CAVP #: A4660, A2624 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.2.8 FCS_COP.1/Hash

Item	Data
Test Assurance Activity	The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented testmacs. The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP. Short Messages Test - Bit-oriented Mode The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. Short Messages Test - Byte-oriented Mode The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. Selected Long Messages Test - Bit-oriented Mode The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. Selected Long Messages Test - Byte-oriented Mode The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. Pseudorandomly Generated Messages Test This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.
Pass/Fail with Explanation	Algorithm: SHA-1, SHA-256, SHA-384, SHA-512 CAVP #: A4660, A2624 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.2.9 FCS_COP.1/KeyedHash

Item	Data
Test Assurance Activity	For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and message data using a known good implementation.
Pass/Fail with Explanation	Algorithm: HMAC (SHA-256, SHA-384, SHA-512) CAVP #: A4660, A2624 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.2.10 FCS_RBG_EXT.1

Item	Data
Test Assurance Activity	The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A). If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call. The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator. Entropy input: the length of the entropy input value must equal the seed length. Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length. Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.

	Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.
Pass/Fail with Explanation	Algorithm: CTR DRBG Mode: AES-256 CAVP #: A4660, A2624 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.3 Auth

7.2.3.1 FCS_CKM.2 RSA

Item	Data
Test Assurance Activity	**Key Establishment Schemes** 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.
Test Steps	This test has been successfully tested in FTP_TRP.1/Admin and FTP_ITC.1 because in both SFRs, evaluator has tested each protocol and verified the successful connection.
Pass/Fail with Explanation	Pass. This test has been successfully tested in FTP_TRP.1/Admin and FTP_ITC.1 because in both SFRs, evaluator has tested each protocol and verified the successful connection.

7.2.3.2 FCS_CKM.2 FCC

Item	Data
Test Assurance Activity	**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.
Pass/Fail with Explanation	NA. Not claimed.

7.2.3.3 FIA_AFL.1 Test#1

Item	Data
Test Assurance Activity	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. ***TD0570 is applied.***
Test Steps	WebUI: • Configure account lockout.

	• Try to connect to the TOE with wrong credentials three consecutive times to lockout the account. • Login with correct credentials and verify that it is not successful. • Verify with logs. SSH: • Configure account lockout. • Try to connect to the TOE with wrong credentials three consecutive times to lockout the account. • Login with correct credentials and verify that it is not successful. • Verify with logs.
Expected Test Results	**The maximum number of successive unsuccessful attempts can be configured on the TOE. The TOE does not allow for access to the device even with correct credentials after an account fails authentication successively for the configured maximum number of unsuccessful attempts.**
Pass/Fail with Explanation	Pass. The TOE did not allow SSH and Web GUI access after using incorrect credentials three times even when using correct credentials.

7.2.3.4 FIA_AFL.1 Test#2a

Item	Data
Test Assurance Activity	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 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). ***TD0570 is applied.***
Pass/Fail with Explanation	NA. Manually unlocking user accounts is not available for the manager device.

7.2.3.5 FIA_AFL.1 Test#2b

Item	Data
Test Assurance Activity	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 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows: 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.
Test Steps	WebUI: • Set user unlock time on the TOE.

	• Attempt to login with incorrect password till the account lockout is triggered. • Verify the account is locked with logs. • Attempt to login with correct password just before the lockout period expiry and verify that it fails while account is still locked. • Verify the account is still locked with logs. • Attempt to login with correct password after lockout time is over, and verify it is successful. • Verify successful login with logs. SSH: • Set user unlock time on the TOE. • Attempt to login with incorrect password till the account lockout is triggered. • Verify the account is locked with logs. • Attempt to login with correct password just before the lockout period expiry and verify that it fails while account is still locked. • Verify the account is still locked with logs. • Attempt to login with correct password after lockout time is over, and verify it is successful. • Verify successful login with logs.
Expected Test Results	• **The TOE should not allow a locked-out user to log in again before the lockout time expires.** • **TOE should show account locked out logs and successful authentication logs once locked out time is completed.**
Pass/Fail with Explanation	Pass. The TOE did not allow a locked-out user to log in again before the lockout time had expired. Once the lock out time had finished, the user was able to login successfully.

7.2.3.6 FIA_PMG_EXT.1 Test#1

Item	Data
Test Assurance Activity	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.
Test Steps	Set minimum password requirements. • Minimum 15-character length. • Minimum 2 upper case letters. • Minimum 2 lower case letters. • Minimum 2 digits. • Minimum 2 special characters. WEBUI: • Create username: good1 password: QWERTYqwerty[] `{|}~12 • Verify with logs that user 'good1' is created. • Create username: good2 password: UIOPASuiopas,:;<=>?\34

	• Verify with logs that user 'good2' is created. • Create username: good3 password: DFGHJKdfghjk*()'+-./56 • Verify with logs that user 'good3' is created. • Create username: good4 password: LZXCVBNMlzxcvbnm!@#$%^&_7890 • Verify with logs that user 'good4' is created. SSH: • Create user with username: good1. • Set 'good1' user's password: AB1CD7E!a@bc1de • Verify with logs the user was created and the password for that user was changed. • Create user with username: good2. • Set 'good2' user's password: FG2HI8J#f$gh2ij • Verify with logs the user was created and the password for that user was changed. • Create user with username: good3. • Set 'good3' user's password: KL3MN9O%k^lm3no • Verify with logs the user was created and the password for that user was changed. • Create user with username: good4. • Set 'good4' user's password: PQ4RS0T&p*qr4st • Verify with logs the user was created and the password for that user was changed. • Create user with username: good5. • Set 'good5' user's password: UV5WX1Y(u)vw5xy • Verify with logs the user was created and the password for that user was changed.
Expected Test Results	**The TOE accepts valid password combinations that meet the requirements on WebUI. Audit logs show that the user with the valid password combination has been added successfully.**
Pass/Fail with Explanation	Pass. The TOE successfully creates user accounts with strong passwords. This meets the testing requirements.

7.2.3.7 FIA_PMG_EXT.1 Test#2

Item	Data
Test Assurance Activity	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.
Test Steps	• Attempt to create a user with a missing upper case character in the password with username: bad & password: ab1cd7e!a@bc1de • Confirm that the user could not be created. • Attempt to create a user with missing lowing case character in password with username: bad1 & password: FG2HI8J#F$GH2IJ • Confirm that the user could not be created.

	• Attempt to create a user with missing digits in the password with username: bad2 & password: KLmMNra%k^lmsno • Confirm that the user could not be created. • Attempt to create a user with a missing special character in the password with username: bad3 & password: PQ4RS0T2prqr4st • Confirm that the user could not be created. • Attempt to create a user with less than 15 characters in password username: bad4 & password: UV5WX1Y(u)vw • Confirm that the user could not be created.
Expected Test Results	**The TOE rejects bad/invalid combinations and only accepts valid password combinations. Audit logs show that addition of users with bad password combinations result in failure due to Invalid Password.**
Pass/Fail with Explanation	Pass. User accounts cannot be created without configured password requirements being met. This meets the testing requirements.

7.2.3.8 FIA_UIA_EXT.1 Test#1

Item	Data
Test Assurance Activity	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.
Test Steps	Console: • Attempt to login local connection with incorrect credentials. • Confirm that access was denied with logs. • Log into the TOE local connection with correct credentials. • Confirm that access was granted with logs. SSH: • Attempt to login remote CLI connection with incorrect credentials. • Confirm that access was denied through logs. • Log into the TOE remote CLI connection with correct credentials. • Confirm that access was granted through logs. WebUI: • Attempt to login remote GUI connection with incorrect credentials. • Confirm that access was denied through logs. • Log into the TOE remote GUI connection with correct credentials. • Confirm that access was granted through logs.
Expected Test Results	**The TOE only allows an authorized user to gain access to the system via console and SSH. Users with incorrect credentials are denied access as shown by audit logs generated.**
Pass/Fail with Explanation	Pass. The TOE only allows an authorized user to gain access to the system via console, SSH, and WebGUI.

7.2.3.9 FIA_UIA_EXT.1 Test#2

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: Test 2: The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.
Test Steps	SSH: • At the SSH login, verify that no functionality except those specified in the requirement is allowed (logon banner). • Verify the logs. WebGUI: • At the GUI login, verify that no functionality except those specified in the requirement is allowed (logon banner). • Verify the logs.
Expected Test Results	**No services except displaying a banner is available to a remote administrator attempting to login to the TOE via SSH or Console.**
Pass/Fail with Explanation	Pass. No system services except the logon banner is available to an unauthenticated user connecting remotely.

7.2.3.10 FIA_UIA_EXT.1 Test#3

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.
Test Steps	• Before logging in via local console, attempt to execute authenticated commands such as show, show auditlog, and show status. This will fail. • Verify with logs.
Expected Test Results	• **No services except displaying a banner is available to a remote administrator attempting to login to the TOE via console.**
Pass/Fail with Explanation	Pass. No services except displaying a banner is available to a remote administrator attempting to login to the TOE via console.

7.2.3.11 FIA_UIA_EXT.1 Test#4

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.
Pass/Fail with Explanation	Pass. The Sensor performs the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2. Therefore, no additional testing is required for this activity. This meets the testing requirements.

7.2.3.12 FIA_UIA_EXT.7 Test#1

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each method of local login allowed: 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.
Test Steps	• At the directly connected login prompt, enter incorrect authentication credentials. Verify that at most obscured feedback is provided. • Verify the failure with logs. • At the directly connected login prompt, enter correct authentication credentials. Verify that at most obscured feedback is provided. • Verify the successful login with logs.
Expected Test Results	**The TOE should not provide anything other than obscured feedback, i.e 'Incorrect Password' when entered credentials are incorrect and no feedback with correct credentials while entering authenticating information.**
Pass/Fail with Explanation	Pass. The TOE only provided obscured feedback when using incorrect credentials and provided nothing when using correct credentials.

7.2.3.13 FMT_MOF.1/ManualUpdate Test#1

Item	Data
Test Assurance Activity	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.
Test Steps	SSH: • Create a user without Security Administrator privileges. • Login as the newly created low privileged user. • Attempt to update the device. This will fail as the required options are unavailable. • Verify via logs that the attempt was unsuccessful. Console:

	• Login as the newly created low privileged user. • Attempt to update the device. This will fail as the required options are unavailable. • Verify via logs that the attempt was unsuccessful.
Expected Test Results	• **Updates will not be available for users without Security Administrative access.** • **Updates - screenshot showing options are disabled.**
Pass/Fail with Explanation	Pass. The high privilege functions are blocked for users with low privileges. This meets the testing requirements.

7.2.3.14 FMT_MOF.1/ManualUpdate Test#2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. This test has been completed as part of the requirements specified in FPT_TUD_EXT.1 Test#1. This meets the testing requirements.

7.2.3.15 FMT_SMF.1 Test#1

Item	Data
Test Assurance Activity	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. **TD0631 has been applied**

Test Steps	The TSF shall be capable of performing the following management functions:
	• Ability to administer the TOE locally and remotely;
	• Ability to configure the access banner;
	• Ability to configure the session inactivity time before session termination or locking;
	• Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;
	• Ability to configure the authentication failure parameters for FIA_AFL.1;
	• [
	  ○ Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);
	  ○ Ability to modify the behaviour of the transmission of audit data to an external IT entity;
	  ○ Ability to configure the cryptographic functionality;
	  ○ Ability to configure the interaction between TOE components;
	  ○ Ability to set the time which is used for time-stamps;
	  ○ Ability to import X.509v3 certificates to the TOE's trust store;
	  ○ Ability to manage the trusted public keys database;]

Expected Test Results	**All management functions identified in Security Target should be met by presenting correct test cases.**

Pass/Fail with Explanation	Pass. All management functions identified have been tested throughout the evaluation. Thus, this requirement has been met.

7.2.3.16 FMT_SMR.2 Test#1

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. There are three interfaces where these can be tested (HTTPS GUI over TLS, console and Remote CLI over SSH) and all test cases use these interfaces. The evaluator has met this requirement through execution of the entirety of this test report by performing actions via all three interfaces.

7.2.3.17 FTA_SSL.3 Test#1

Item	Data
Test Assurance Activity	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.
Test Steps	WebUI: For 5 minutes: • Configure a remote GUI time out period of 5 minutes on administrative sessions. • Connect to the TOE from the remote GUI and record the login time. • Let the GUI sit idle for the mentioned time and verify that the session was terminated. • Verify that the session is terminated with logs. WebUI: For 10 minutes: • Configure a remote GUI time out period of 10 minutes on administrative sessions. • Connect to the TOE from the remote GUI and record the login time. • Let the GUI sit idle for the mentioned time and verify that the session was terminated. • Verify that the session is terminated with logs. SSH: For 2 minutes: • Configure a time out period of 2 minutes on administrative sessions. • Connect to the TOE from the remote CLI and let the remote CLI connection be idle until the session is terminated.

	• Verify that the session is terminated with logs. SSH: For 5 minutes: • Configure a time out period of 5 minutes on administrative sessions. • Connect to the TOE from the remote CLI and let the remote CLI connection be idle until the session is terminated. • Verify that the session is terminated with logs.
Expected Test Results	**The TOE should terminate idle remote sessions after the specified time.**
Pass/Fail with Explanation	Pass. The TOE terminated the idle remote sessions after the specified time.

7.2.3.18 FTA_SSL.4 Test#1

Item	Data
Test Assurance Activity	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.
Test Steps	• Log onto the TOE through a local administrative interface. • Using the instructions provided by the user guide log off. • Verify the logs reflect the log off
Expected Test Results	• **Evidence (e.g., screenshot or CLI output) from logging into the TOE locally.** • **Evidence (e.g., screenshot or CLI output) showing the log out.** • **Log showing the log out.**
Pass/Fail with Explanation	Pass. The TOE correctly terminated the remote console sessions.

7.2.3.19 FTA_SSL.4 Test#2

Item	Data
Test Assurance Activity	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.
Test Steps	WebUI: • Log onto the TOE through a remote CLI interface. • Using the instructions provided by the user guide log off. • Verify the logs reflect the log out. SSH:

	• Log onto the TOE through a remote CLI interface. • Using the instructions provided by the user guide log off. • Verify the logs reflect the log out.
Expected Test Results	**The TOE should allow users to terminate the remote sessions. Audit logs show the successful login and logout of user from TOE.**
Pass/Fail with Explanation	Pass. The TOE allows users to terminate the remote sessions. Audit logs show the successful login and logout of the user from the TOE.

7.2.3.20 FTA_SSL_EXT.1.1 Test#1

Item	Data
Test Assurance Activity	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 reauthentication is needed when trying to unlock the session.
Test Steps	• Configure a local CLI time out period of 1 minute on administrative sessions. • Connect to the TOE from the console. • Let the local CLI connection be idle for 1 minute. Verify that the session is terminated. • Verify that the session is terminated with logs. • Configure a local CLI time out period of 2 minutes on administrative sessions. • Connect to the TOE from the console. • Let the local CLI connection be idle for 2 minutes. Verify that the session is terminated. • Verify that the session is terminated with logs.
Expected Test Results	**The TOE should terminate idle local sessions after the specified time.**
Pass/Fail with Explanation	Pass. The TOE terminated the idle local session after the specified time.

7.2.3.21 FTA_TAB.1 Test#1

Item	Data
Test Assurance Activity	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.
Test Steps	Console: • Login to the TOE using console.

	• Configure remote login banner on the TOE. • Log off and login again to verify that the banner is being displayed. SSH: • Login to the TOE using SSH. • Configure remote login banner on the TOE. • Log off and login again to verify that the banner is being displayed. WEBUI: • Login to the TOE via WEBUI and configure the banner. • Logoff and login again and verify that banner is being displayed.
Expected Test Results	**When any user accesses the TOE through the SSH or GUI, the configured banner should be displayed prior to authenticating the TOE.**
Pass/Fail with Explanation	Pass. When any user accesses the TOE through the console, SSH and GUI, the configured banner is displayed prior to authenticating the TOE.

7.2.3.22 FTP_TRP.1/Admin Test#1

Item	Data
Test Assurance Activity	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.
Test Steps	SSH: • Start an administrative session with the device. • Capture the packets between the remote workstation and the TOE and verify that the connection is successful. • Verify the successful connection via logs. WebGUI: • Start an administrative session with the device. • Capture the packets between the remote workstation and the TOE and verify that the connection is successful. • Verify the successful connection via logs.
Expected Test Results	• **Successful communication between TOE and remote administrator via SSH.** • **Flow of application data packets in TLS/HTTPS connection in confirms successful connection with the TOE.**
Pass/Fail with Explanation	Pass. The TOE is successfully able to communicate with other devices via each remote access method using encrypted traffic.

7.2.3.23 FTP_TRP.1/Admin Test#2

Item	Data
Test Assurance Activity	The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FTP_TRP.1/Admin Test#1 and FCS_HTTPS_EXT.1. The TOE does not transmit any data in plaintext and only sends encrypted traffic.

7.2.3.24 FCS_HTTPS_EXT.1

Item	Data
Test Assurance Activity	This test is now performed as part of FIA_X509_EXT.1/Rev testing. Tests are performed in conjunction with the TLS evaluation activities. If the TOE is an HTTPS client or an HTTPS server utilizing X.509 client authentication, then the certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FTP_TRP.1/Admin Test#1. The TOE does not transmit any data in plaintext and only sends encrypted traffic.

7.2.4 Distributed

7.2.4.1 FAU_GEN.1 Test#2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. The audit records associated with each test case are recorded with each test case, in the test report for each component. A comparison of required audit records to the presented audit records was additionally performed and is included in FAU_GEN.1 Test #1. This analysis shows that each required audit record is generated by each TOE component according to the allocation of SFRs, meeting the test requirements.

7.2.4.2 FAU_GEN.2 Test#1

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FCO_CPC_EXT.1 Test#1.1.

7.2.4.3 FAU_STG_EXT.4 Test#1

Item	Data
Test Assurance Activity	For each type of TOE component, the evaluator shall perform a representative subset of auditable actions and ensure that these actions cause the generation of appropriately formed audit records. Generation of such records can be observed directly on the distributed TOE component (if there is appropriate interface), or indirectly after transmission to a central location.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with the 7500 sensor's FAU_STG_EXT.5 Test #1.

7.2.4.4 FAU_STG_EXT.4 Test#2

Item	Data
Test Assurance Activity	For each type of TOE component that, in the evaluated configuration, is capable of transmitting audit information to the external audit server (as specified in FTP_ITC.1), the evaluator shall configure a trusted channel and confirm that audit records generated as a result of actions taken by the evaluator are securely transmitted. It is sufficient to observe negotiation and establishment of the secure channel with the TOE component and the subsequent transmission of encrypted data to confirm this functionality. Alternatively, the following steps shall be performed: The evaluator induces audit record transmission, then reviews the packet capture around the time of transmission and verifies that no audit data is transmitted in the clear.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with the 7500 sensor's FAU_STG_EXT.5 Test #1.

7.2.4.5 FAU_STG_EXT.4 Test#3

Item	Data
Test Assurance Activity	For each type of TOE component that, in the evaluated configuration, is capable of transmitting audit information to another TOE component (as specified in FTP_ITT.1 or FTP_ITC.1, respectively), the evaluator shall configure a secure channel and confirm that audit records generated as a result of actions taken by the evaluator are securely transmitted. It is sufficient to observe negotiation and establishment of the secure channel with the TOE component and the subsequent

	transmission of encrypted data to confirm this functionality. Alternatively, the following steps shall be performed: The evaluator induces audit record transmission, then reviews the packet capture around the time of transmission and verifies that no audit data is transmitted in the clear.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with the 7500 sensor's FAU_STG_EXT.5 Test #1.

7.2.4.6 FCO_CPC_EXT.1 Test#1.1

Item	Data
Test Assurance Activity	Test 1.1: the evaluator shall confirm that an IT entity that is not currently a member of the distributed TOE cannot communicate with any component of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE components that it is required to communicate with (non-equivalent TOE components are as defined in the minimum configuration for the distributed TOE)
Test Steps	• Check the device manager in the manager GUI for configured sensors. • Configure the sensor to communicate with the manager without adding it to the manager configuration. • Verify that the sensor does not get added as a configured device in the device manager. • Verify the trust establishment failure via the manager logs. • Verify via packet capture that the sensor attempts trust establishment with the manager and fails to do so. • Configure the manager to communicate with the sensor by adding the latter in the device manager. • Configure the corresponding sensor to communicate with the manager. • Verify that the sensor gets added as a configured device in the device manager. • Verify the trust establishment success via the manager logs. • Verify via packet capture that the configured sensor attempts trust establishment with the manager and succeeds.
Expected Test Results	**The TOE cannot communicate with any component of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE components.**
Pass/Fail with Explanation	Pass. The TOE cannot communicate with any component of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE components.

7.2.4.7 FCO_CPC_EXT.1 Test#1.2

Item	Data
Test Assurance Activity	Test 1.2: the evaluator shall confirm that after enablement, an IT entity can communicate only with the components that it has been enabled for. This includes testing that the enabled communication is successful for the enabled component pair, and that communication remains unsuccessful with any other component for which communication has not been explicitly enabled

	Some TOEs may set up the registration channel before the enablement step is carried out, but in such a case the channel must not allow communications until after the enablement step has been completed.
Test Steps	• Verify the status of the enabled NS7500 sensor in the device manager GUI. • Verify via logs that the sensor is enabled successfully. • Verify via packet capture the successful communication between the manager and the enabled sensor. • While this sensor is enabled, initiate communication from another non-enabled sensor. • Verify that the non-enabled sensor does not get added as a configured device in the device manager. • Verify the trust establishment failure via the manager logs. • Verify via packet capture that the non-enabled sensor attempts trust establishment with the manager and fails to do so.
Expected Test Results	• **TOE communication with an enabled component is successful.** • **TOE communication with a component that hasn't been explicitly enabled is unsuccessful.**
Pass/Fail with Explanation	Pass. TOE communication with an enabled component is successful while the same with a component that hasn't been explicitly enabled is unsuccessful.

7.2.4.8 FCO_CPC_EXT.1 Test#2

Item	Data
Test Assurance Activity	Test 2: The evaluator shall separately disable each TOE component in turn and ensure that the other TOE components cannot then communicate with the disabled component, whether by attempting to initiate communications with the disabled component or by responding to communication attempts from the disabled component.
Test Steps	• Verify the status of the enabled NS7500 sensor in the device manager GUI. • Disconnect the sensor from the manager. • Verify via device manager that the sensor is removed from the manager. • Verify via logs that the sensor is removed from the manager. • Initiate communication from the now disabled sensor. • Verify that the disabled sensor does not get added as a configured device in the device manager. • Verify the trust establishment failure via the manager logs. • Verify via packet capture that the disabled sensor attempts trust establishment with the manager and fails to do so.
Expected Test Results	**Disabled components are unable to successfully communicate with the TOE.**
Pass/Fail with Explanation	Pass. Disabled components are unable to successfully communicate with the TOE.

7.2.4.9 FCO_CPC_EXT.1 Test#3

Item	Data
Test Assurance Activity	Test 3: The evaluator shall carry out the following tests according to those that apply to the values of the main (outer) selection made in the ST for FCO_CPC_EXT.1.2. 1) If the ST uses the first type of communication channel in the selection in FCO_CPC_EXT.1.2 then the evaluator tests the channel via the Evaluation Activities for FTP_ITC.1 or FPT_ITT.1 according to the second selection – the evaluator shall ensure that the test coverage for these SFRs includes their use in the registration process. 2) If the ST uses the second type of communication channel in the selection in FCO_CPC_EXT.1.2 then the evaluator tests the channel via the Evaluation Activities for FTP_TRP.1/Join. 3) If the ST uses the 'no channel' selection, then no test is required.
Pass/Fail with Explanation	Pass. Test covered by FPT_ITT.1 (the ST uses the first type of communications channel).

7.2.4.10 FCO_CPC_EXT.1 Test#4

Item	Data
Test Assurance Activity	Test 4: The evaluator shall perform one of the following tests, according to the TOE characteristics identified in its TSS and operational guidance: 1) If the registration channel is not subsequently used for intercomponent communication, and in all cases where the second selection in FCO_CPC_EXT.1.2 is made (i.e. using FTP_TRP.1/Join) then the evaluator shall confirm that the registration channel can no longer be used after the registration process has completed, by attempting to use the channel to communicate with each of the endpoints after registration has completed 2) If the registration channel is subsequently used for intercomponent communication then the evaluator shall confirm that any aspects identified in the operational guidance as necessary to meet the requirements for a steady-state intercomponent channel (as in FTP_ITC.1 or FPT_ITT.1) can indeed be carried out (e.g. there might be a requirement to replace the default key pair and/or public key certificate).
Pass/Fail with Explanation	Pass. Test covered by FTP_ITT.1 (the registration channel is subsequently used for intercomponent communication)

7.2.4.11 FCO_CPC_EXT.1 Test#5

Item	Data
Test Assurance Activity	For each aspect of the security of the registration channel that operational guidance states can be modified by the operational environment in order to improve the channel security (cf. AGD_PRE.1

	refinement item 2 in (cf. the requirements on Preparative Procedures in 3.5.1.2), the evaluator shall confirm, by following the procedure described in the operational guidance, that this modification can be successfully carried out.
Pass/Fail with Explanation	Pass. Test covered by FCO_CPC_EXT.1 Test #1.1 (the PSK is configured when the sensor is initialized).

7.2.4.12 FPT_ITT Test#1

Item	Data
Test Assurance Activity	Test 1: The evaluator shall ensure that communications using each protocol between each pair of authorized TOE components is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.
Pass/Fail with Explanation	Pass. This test was performed in conjunction with FCO_CPC_EXT.1 Test#1.1. External connections from the TOE to other authorized TOE components are sent via an encrypted channel.

7.2.4.13 FPT_ITT Test#2

Item	Data
Test Assurance Activity	Test 2: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
Pass/Fail with Explanation	Pass. This test was performed in conjunction with FAU_STG_EXT.4 Test #1 and FTP_ITC.1 Test #4. External connections from the TOE are sent via an encrypted channel.

7.2.4.14 FPT_ITT Test#3

Item	Data
Test Assurance Activity	Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route between distributed components. The evaluator shall ensure that, for each different pair of non-equivalent component types, the connection is physically interrupted for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) a duration that is shorter than the application layer timeout but is of sufficient length to interrupt the network link layer. The evaluator shall ensure that when physical connectivity is restored, either communications are appropriately protected, or the secure channel is terminated and the registration process (as described in the FTP_TRP.1/Join) re-initiated, with the TOE generating adequate warnings to alert the Security Administrator. In the case that 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 components. The interruption shall not be performed at the virtual node (e.g. virtual switch) and must be physical in nature.
Test Steps	• Verify that the sensor and the manager are connected. • Disconnect the cable between the sensor and Ethernet switch for less than the application layer timeout of 60 seconds and verify via PCAPs that the TLS session was not

	dropped, and that all data continues to be encrypted. The existing TLS connection was maintained, and all data sent between the systems is encrypted. • Disconnect the cable between the sensor and Ethernet switch more than the application layer timeout of 60 seconds and verify via PCAPs that the TLS session was re-established, and that all data is encrypted. The existing TLS connection was dropped, and a new connection was established before any user data was transmitted.
Expected Test Results	• **The SSH and TLS connection should be encrypted once the physical connection restores within the application layer timeout and if it exceeded the application layer time.** • **Verify in packet capture.**
Pass/Fail with Explanation	Pass. The TLS connections were maintained or re-established as necessary, and all data sent between the system was encrypted. This satisfies the testing requirements.

7.2.4.15 FTP_TRP.1/Join Test#1

Item	Data
Test Assurance Activity	The evaluator shall ensure that the communications path for joining components to the TSF is tested for each distinct (non-equivalent) component type, setting up the connections as described in the guidance documentation and ensuring that communication is successful. In particular the evaluator shall confirm that requirements on environment protection for the registration process are consistent with observations made on the test configuration (for example, a requirement to isolate the components from the Internet during registration might be inconsistent with the need for a component to contact a license server). If no requirements on the registration environment are identified as necessary to protect confidentiality, then the evaluator shall confirm that the key used for registration can be configured (following the instructions in the guidance documentation) to be at least the same length as the key used for the internal TSF channel that is being enabled. The evaluator shall confirm that the key used for the channel is unique to the pair of components (this is done by identifying the relevant key during the registration test: it is not necessary to examine the key value). *The intention here is to cover all different software sections involved. For example, a single software image may be installed on different TOE components, but with different sections of the image executed according to the hardware platform or communications stack. In such as case tests should be carried out for each different software section
Pass/Fail with Explanation	N/A. This SFR is not claimed.

7.2.4.16 FTP_TRP.1/Join Test#2

Item	Data
Test Assurance Activity	The evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be enabled by a Security Administrator for all the TOE components identified in the guidance documentation as capable of initiation.
Pass/Fail with Explanation	N/A. This SFR is not claimed.

7.2.4.17 FTP_TRP.1/Join Test#3

Item	Data
Test Assurance Activity	The evaluator shall ensure that if the guidance documentation states that the channel data is encrypted then the data observed on the channel is not plaintext.
Pass/Fail with Explanation	N/A. This SFR is not claimed.

7.2.5 SSHS

7.2.5.1 FCS_SSHS_EXT.1.2 Test#1

Item	Data
Test Assurance Activity	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. **TD0631 has been applied.**
Test Steps	• Enable public key-based authentication on the TOE. <u>SSH-RSA:</u> • Generate a public key using SSH-RSA algorithm and note its fingerprint. • Copy the generated SSH-RSA public key on the TOE. • Log into the TOE with public key-based authentication. • Verify the successful connection using TOE logs. • Verify the successful connection via packet capture. <u>RSA-SHA2-256:</u> • Generate a public key using RSA-SHA2-256 algorithm and note its fingerprint. • Copy the generated RSA-SHA2-256 public key on the TOE. • Log into the TOE with public key-based authentication. • Verify the successful connection using TOE logs.

	• Verify the successful connection via packet capture. RSA-SHA2-512: • Generate a public key using RSA-SHA2-512 algorithm and note its fingerprint. • Copy the generated RSA-SHA2-512 public key on the TOE. • Log into the TOE with public key-based authentication. • Verify the successful connection using TOE logs. • Verify the successful connection via packet capture. ECDSA-SHA2-NISTP256: • Generate a public key using ECDSA-SHA2-NISTP256 algorithm and note its fingerprint. • Copy the generated ECDSA-SHA2-NISTP256 public key on the TOE. • Log into the TOE with public key-based authentication. • Verify the successful connection using TOE logs. • Verify the successful connection via packet capture.
Expected Test Results	**The TOE must successfully establish a SSH session connection with the client using public key authentication.**
Pass/Fail with Explanation	Pass. The TOE is able establish a SSH session connection with the client successfully using the supported public key algorithms.

7.2.5.2 FCS_SSHS_EXT.1.2 Test#2

Item	Data
Test Assurance Activity	Test objective: The purpose of these tests is to verify server supports each claimed client authentication method. 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. **TD0631 has been applied.**
Test Steps	• Verify the key configured on the TOE. • Configure the SSH client with a new ECDSA keypair for SSH and do not configure the TOE to recognize the client's public key. • Log into the TOE via SSH using ecdsa-256-based authentication. • Verify failed authentication logs on TOE. • Verify authentication failure via packet capture.
Expected Test Results	• **The TOE should reject SSH connections when incorrect/unknown public keys are presented.**

	• **When a connection is attempted using an ecdsa-sha2-nistp256 public key value that has not been loaded onto the TOE, public key authentication fails, and the user is prompted for a password.**
Pass/Fail with Explanation	Pass. The TOE does not allow a connection when an invalid password is entered. This meets the testing requirement.

7.2.5.3 FCS_SSHS_EXT.1.2 Test#3

Item	Data
Test Assurance Activity	Test objective: The purpose of these tests is to verify server supports each claimed client authentication method. 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. **TD0631 has been applied.**
Test Steps	• Create a user on the TOE and set a password. • Log into the TOE via SSH with password authentication using correct credentials. • Verify using authentication logs on TOE. • Verify via packet capture that SSH session was established.
Expected Test Results	• **The TOE should set up a user with password-based authentication.** • **User authentication succeeds when the correct password is provided by the user.**
Pass/Fail with Explanation	Pass. The TOE accepts password-based authentication from a remote SSH client when the correct password is provided. This meets the testing requirements.

7.2.5.4 FCS_SSHS_EXT.1.2 Test#4

Item	Data
Test Assurance Activity	Test objective: The purpose of these tests is to verify server supports each claimed client authentication method. 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. **TD0631 has been applied.**
Test Steps	• Create a user on the TOE and set a password.

	• Attempt to Log into the TOE via SSH with correct username and incorrect password and observe the connection getting rejected by the TOE. • Verify authentication via logs that reflect failures. • Verify authentication via packet capture that reflect failures.
Expected Test Results	• **The TOE should set up a user with password-based authentication.** • **User authentication should fail when incorrect password is provided by the user.**
Pass/Fail with Explanation	Pass.The TOE does not establish a connection with a remote SSH user when incorrect password are presented. This meets the testing requirements.

7.2.5.5 FCS_SSHS_EXT.1.3 Test#1

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.
Test Steps	• Establish an SSH connection to the TOE, using the acumen-sshs tool to send a packet larger than the established limit. • Verify connection failure via logs. • Verify session disconnection due to large packets via packet capture.
Expected Test Results	• **The TOE must drop a packet larger than the allowed limit.** • **Logs show the TOE dropping the connection when a packet sent is larger than allowed range.**
Pass/Fail with Explanation	Pass. The TOE drops packets larger than the specified limit that are received within an SSH session. This meets the testing requirements.

7.2.5.6 FCS_SSHS_EXT.1.4 Test#1

Item	Data
Test Assurance Activity	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.

Test Steps	• Connect to the TOE using aes128-gcm@openssh.com encryption. • Verify that the SSH session was encrypted using aes128-gcm@openssh.com via log. • Verify that the SSH session was encrypted using aes128-gcm@openssh.com via packet capture. • Connect to the TOE using aes256-gcm@openssh.com encryption. • Verify that the SSH session was encrypted using aes256-gcm@openssh.com via log. • Verify that the SSH session was encrypted using aes256-gcm@openssh.com via packet capture. • Connect to the TOE using an unsupported encryption algorithm (AES128-CBC). • Verify that the SSH session was refused via log. • Verify that the SSH session was refused via packet capture.
Expected Test Results	• **The TOE should establish a SSH session only with the claimed encryption algorithms.** • **The connection must fail in case of unsupported encryption algorithms used.**
Pass/Fail with Explanation	Pass. The TOE is able to establish a SSH session with the client successfully using only the claimed encryption algorithms. This meets the testing requirements.

7.2.5.7 FCS_SSHS_EXT.1.5 Test#1

Item	Data
Test Assurance Activity	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 algorithms. It is sufficient to observe (on the wire) the successful negotiation of the algorithms to satisfy the intent of the test. **TD0631 has been applied.**
Test Steps	• Verify that the claimed hostkey algorithms are supported by the TOE. <u>SSH-RSA:</u> • Generate an ssh-rsa host key pair on the TOE and note its fingerprint. • Login to the TOE using the host public key and verify that the session is established. • Verify via logs that the session was established using the configured host key algorithm. • Verify via packet capture that the configured host key algorithm was used. <u>RSA-SHA2-256:</u> • Generate an rsa-sha2-256 host key pair on the TOE and note its fingerprint. • Login to the TOE using the host public key and verify that the session is established. • Verify via logs that the session was established using the configured host key algorithm.

	• Verify via packet capture that the configured host key algorithm was used. RSA-SHA2-512: • Generate an rsa-sha2-512 host key pair on the TOE and note its fingerprint. • Login to the TOE using the host public key and verify that the session is established. • Verify via logs that the session was established using the configured host key algorithm. • Verify via packet capture that the configured host key algorithm was used. ECDSA-SHA2-NISTP256: • Generate an ecdsa-sha2-nistp256 host key pair on the TOE and note its fingerprint. • Login to the TOE using the host public key and verify that the session is established. • Verify via logs that the session was established using the configured host key algorithm. • Verify via packet capture that the configured host key algorithm was used.
Expected Test Results	**TOE establishes a successful SSH connection only with the claimed host key algorithms.**
Pass/Fail with Explanation	Pass. The TOE establishes a successful SSH connection using each one of the claimed host public key algorithms.

7.2.5.8 FCS_SSHS_EXT.1.5 Test#2

Item	Data
Test Assurance Activity	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. **TD0631 has been applied.**
Test Steps	• Attempt to establish a SSH session using the SSH-DSS host public key algorithm and verify that it fails. • Verify that the SSH session was refused via log. • Verify that the connection is refused via packet capture.
Expected Test Results	**TOE should reject a connection request from unclaimed host public key algorithm.**
Pass/Fail with Explanation	Pass. The TOE refuses a SSH connection when a non-supported host public key algorithm is used. This meets the test requirements.

7.2.5.9 FCS_SSHS_EXT.1.6 Test#1

Item	Data

Test Assurance Activity	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.
Pass/Fail with Explanation	NA. The connection uses an implicit MAC. Therefore, this test is not applicable.

7.2.5.10 FCS_SSHS_EXT.1.6 Test#2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	NA. The connection uses an implicit MAC. Therefore, this test is not applicable.

7.2.5.11 FCS_SSHS_EXT.1.7 Test#1

Item	Data
Test Assurance Activity	The evaluator shall configure an SSH client to only allow the diffie-hellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.
Test Steps	• Attempt to establish a connection with the TOE from an SSH client using Diffie-hellman-group1-sha1 as the key exchange method and verify that it fails. • Verify connection failure via packet capture. • Verify that the session was not established via logs generated on the TOE.
Expected Test Results	• **The TOE does not permit connections when using diffiehellman-group1-sha1. User gets a 'no matching key exchange method found' error.** • **Packet capture should show the SSH connection getting rejected by the TOE when unsupported kex-algorithm is used by the SSH client.** • **Audit logs should reflect failure of connection.**
Pass/Fail with Explanation	Pass. The SSH connection fails when Diffie-hellman-group1-sha1 (a non-approved algorithm) for the key exchange is used. This meets the testing requirement.

7.2.5.12 FCS_SSHS_EXT.1.7 Test#2

Item	Data
Test Assurance Activity	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.
Test Steps	• Attempt to establish a connection with the TOE from an SSH client using ecdh-sha2-nistp256 as the key exchange method and verify that it succeeds. • Verify the successful connection using packet capture. • Verify that the session was established via TOE generated logs.
Expected Test Results	• **The TOE should make a SSH connection using ecdh-sha2-nistp256 key exchange algorithm.** • **Encrypted packet should flow between the two devices and the packet capture verifies successful SSH connection when using a supported key exchange algorithm.** • **Audit logs shows successful connection.**
Pass/Fail with Explanation	Pass. The TOE is able to make SSH connections with the claimed key exchange method. This meets the testing requirements.

7.2.5.13 FCS_SSHS_EXT.1.8 Test#1a

Item	Data
Test Assurance Activity	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. 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).
Test Steps	• Set the time-based threshold for SSH connections to 3 minutes (180 seconds) on the TOE. • Initiate a new SSH session in verbose mode with the TOE. • Periodically repeat the ‘date’ command until rekeying occurs. • Verify via logs that rekey takes place after the time-based threshold.

Expected Test Results	• **The TOE issues a rekey after the specified time as configured on the TOE.** • **Audit logs show the session rekey request has been sent after a time-based threshold has been reached.**
Pass/Fail with Explanation	Pass. The TOE initiates a rekey after every set interval of 3 minutes. This meets the testing requirement.

7.2.5.14 FCS_SSHS_EXT.1.8 Test#1b

Item	Data
Test Assurance Activity	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 traffic-based threshold the evaluator shall use the TOE to connect to an SSH client and shall transmit data to and/or receive data from the TOE within the active SSH session until the threshold for data protected by either encryption key is reached. It is acceptable if the rekey occurs before the threshold is reached (e.g. because the traffic is counted according to one of the alternatives given in the Application Note for FCS_SSHS_EXT.1.8). The evaluator shall verify that more data has been transmitted within the SSH session than the threshold allows and shall verify that the TOE initiated a rekey (the method of verification shall be reported by the evaluator). Testing does not necessarily have to be performed with the threshold configured at the maximum allowed value of one gigabyte of transferred traffic but the value used for testing shall not exceed one gigabyte. The evaluator needs to ensure that the rekeying has been initiated by the TOE and not by the SSH client that is connected to the TOE. If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, the evaluator needs to verify that the threshold(s) can be configured as described in the guidance documentation and the evaluator needs to test that modification of the thresholds is restricted to Security Administrators (as required by FMT_MOF.1/Functions). In cases where data transfer threshold could not be reached due to hardware limitations it is acceptable to omit testing of this (SSH rekeying based on data transfer threshold) threshold if both the following conditions are met: 1. An argument is present in the TSS section describing this hardware- based limitation and 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.
Test Steps	• Set the SSH traffic-based threshold for TOE to 100 MB. • Initiate a new SSH session on TOE and fetch a file more than the set threshold from a non-TOE entity using sftp in verbose mode. • Verify via logs that rekey takes place after set the data limit.

Expected Test Results	• **The TOE issues a rekey after the specified amount of data is transferred as configured on the TOE.** • **Audit logs show session rekey requests being sent after reaching the set data limit.**
Pass/Fail with Explanation	Pass. The TOE correctly issued a rekey after the data limit had exceeded the set threshold. This meets the testing requirement.

7.2.6 TLSC

7.2.6.1 FCS_TLSC_EXT.1.1 Test#1

Item	Data
Test Assurance Activity	The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	**TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256** • Configure the TOE to connect to the syslog server over TLS. • Establish a TLS connection using the TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ciphersuite • Verify that the session was established with the chosen ciphersuite **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384** • Establish a TLS connection using the TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ciphersuite Verify that the session was established with the chosen ciphersuite
Expected Test Results	**TOE logs show the successful establishment of TLS connection.** **Packet Captures show the successful establishment of TLS connection with configured ciphersuites.**
Pass/Fail with Explanation	Pass: The TOE allows a connection with all claimed cipher suites. This meets testing requirements.

7.2.6.2 FCS_TLSC_EXT.1.1 Test#2

Item	Data
Test Assurance Activity	The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field, and

	a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.
Test Steps	• Create a server certificate with the Server Authentication EKU. • Attempt a connection from the TOE to a TLS server using the certificate that contains the Server Authentication EKU. • Verify that the TOE accepts the connection. • Create a server certificate that lacks the Server Authentication EKU. • Attempt a connection from the TOE to a TLS server using the invalid certificate missing the Server Authentication EKU. • Verify that the TOE rejects the connection. • Verify with logs.
Expected Test Results	**TOE should establish a connection with a server with authorized server certificate otherwise TOE should reject the connection.**
Pass/Fail with Explanation	Pass. The TOE does not make the connection because the evaluation of the extendedkeyusage field fails. This meets the testing requirements.

7.2.6.3 FCS_TLSC_EXT.1.1 Test#3

Item	Data
Test Assurance Activity	The evaluator shall send a server certificate in the TLS connection that the does not match the server-selected ciphersuite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite). The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.
Test Steps	• Initiate a TLS connection using acumen-tlsc tool such that the server certificate presented (ECDSA) doesn't match the server-selected ciphersuite (RSA). • Verify that the connection is not established through packet capture. • Verify that a log is a generated indicating that connection was terminated.
Expected Test Results	**The TOE should be unable to establish a connection with non-supported ciphersuite.**
Pass/Fail with Explanation	Pass. The TOE is unable to establish a connection with non-supported ciphersuite.

7.2.6.4 FCS_TLSC_EXT.1.1 Test#4a

Item	Data
Test Assurance Activity	The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection
Test Steps	• Attempt a connection to the server using the TLS_NULL_WITH_NULL_NULL ciphersuite using acumen-tlsc tool and verify that it fails. • Verify that the TOE denies the connection using packet capture.

	• Verify connection failure with logs.
Expected Test Results	**The TOE rejects the server connection with TLS_NULL_WITH_NULL_NULL ciphersuite.**
Pass/Fail with Explanation	Pass. The TOE rejects the server connection with TLS_NULL_WITH_NULL_NULL ciphersuite.

7.2.6.5 FCS_TLSC_EXT.1.1 Test#4b

Item	Data
Test Assurance Activity	Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.
Test Steps	• Attempt a connection from the TOE to a remote TLS server using acumen-tlsc tool to modify the server's ciphersuite to one not present in the Client Hello. Verify that the connection fails. • Verify connection failure with packet capture. • Verify connection failure with logs.
Expected Test Results	**The TOE rejects the connection after receiving the Server Hello packet due to a common ciphersuite not being agreed upon.**
Pass/Fail with Explanation	Pass. The TOE rejects connection after receiving a corrupted server hello packet.

7.2.6.6 FCS_TLSC_EXT.1.1 Test#4c

Item	Data
Test Assurance Activity	[conditional]: If the TOE presents the **Supported Elliptic Curves/Supported Groups Extension** the evaluator shall configure the server to perform an ECDHE or DHE key exchange in the TLS connection using a non-supported curve/group (for example P-192) and shall verify that the TOE disconnects after receiving the server's Key Exchange handshake message.
Test Steps	• Use 'acumen-tlsc' tool to configure the server to perform a connection with an unsupported curve/group and verify that it fails. • Verify via packet capture that the TOE disconnects after receiving the server's key exchange handshake message. • Verify failure with logs.
Expected Test Results	**The TOE rejects connection after receiving the server's Key Exchange handshake message as a non-supported curve/group is used for key exchange while establishing a tls connection.**
Pass/Fail with Explanation	Pass. The TOE rejects the connection when a non-supported curve/group is used for key exchange while establishing a tls connection.

7.2.6.7 FCS_TLSC_EXT.1.1 Test#5a

Item	Data
Test Assurance Activity	Change the TLS version selected by the server in the Server Hello to a non-supported TLS version and verify that the client rejects the connection.
Test Steps	• Using acumen-tlsc tool, attempt a connection to a remote TLS server using a non-supported TLS version and verify that the TOE rejects the connection. • Verify the connection fails with packet capture. • Verify failure with logs.
Expected Test Results	**The TOE rejects connection due to unsupported TLS version.**
Pass/Fail with Explanation	Pass. The TOE rejects a connection that uses an unsupported TLS version.

7.2.6.8 FCS_TLSC_EXT.1.1 Test#5b

Item	Data
Test Assurance Activity	[conditional]: If **using DHE or ECDH**, modify the signature block in the Server's Key Exchange handshake message, and verify that the handshake does not finished successfully, and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.
Test Steps	• Attempt a connection from the TOE to a remote TLS server using acumen-tlsc tool that would allow the server's signature block to be modified. Verify that the connection fails. • Verify the connection failure with packet capture. • Verify the connection fails with logs.
Expected Test Results	**The TOE rejects the connection, as the server signature block in the Server's Key Exchange handshake message has been modified by a third party and thus the handshake is not finished.**
Pass/Fail with Explanation	Pass. The TOE rejects the connection, as the server signature block in the Server's Key Exchange handshake message has been modified by a third party. The handshake is not finished successfully.

7.2.6.9 FCS_TLSC_EXT.1.1 Test#6a

Item	Data
Test Assurance Activity	Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application data flows.
Test Steps	• Attempt a connection to a modified TLS Server using acumen-tlsc tool to modify a byte in the Server Finished handshake message. Verify that the connection fails. • Verify the failed connection via packet capture. • Verify via logs that the connection fails.
Expected Test Results	**The TOE rejects the connection, and the handshake is not finished as the server finished message is not being sent.**

Pass/Fail with Explanation	Pass. The TOE rejects the connection due to the modified Server Finished message, and the handshake is not finished.

7.2.6.10 FCS_TLSC_EXT.1.1 Test#6b

Item	Data
Test Assurance Activity	Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake does not finish successfully and no application data flows.
Test Steps	• Attempt a connection to a modified TLS server using acumen-tlsc that would allow sending a garbled message from the server after the server issues the ChangeCipherSpec message and verify that the TOE rejects the connection. • Verify failure with packet capture. • Verify failure with logs.
Expected Test Results	**The TOE rejects the connection, as the TOE is not able to decrypt the ChangeCipherSpec packet due to the corruption introduced by acumen-tlsc tool. Thus the handshake is not finished.**
Pass/Fail with Explanation	Pass. The TOE rejects the connection, as the TOE is not able to decrypt the ChangeCipherSpec packet due to the corruption introduced by acumen-tlsc tool. Thus the handshake is not finished.

7.2.6.11 FCS_TLSC_EXT.1.1 Test#6c

Item	Data
Test Assurance Activity	Modify at least one byte in the server's nonce in the Server Hello handshake message and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.
Test Steps	• Attempt a connection to a modified TLS server using acumen-tlsc that would allow sending a modified nonce from the server and verify that the TLS handshake with the TOE fails. • Verify failure with packet capture. • Verify failure with logs.
Expected Test Results	**The TOE rejects the Server Key Exchange handshake message as the server's nonce used for authentication and other cryptographic functions is modified by the acumen-tlsc tool.**
Pass/Fail with Explanation	Pass. The TOE rejects the Server Key Exchange handshake message as the server's nonce used for authentication and other cryptographic functions is modified by the acumen-tlsc tool.

7.2.6.12 FCS_TLSC_EXT.1.2 Test#1

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.**

	The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN. Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1.
Test Steps	CN as IPV4: • Configure the TOE for reference identifier name as IPV4. • Configure the Server certificate showing invalid CN and missing SAN extension. • Establish a TLS connection with the syslog server using above server certificate with acumen-tlsc tool and verify the connection failure. • Verify the connection failure logs on the device. • Verify the unsuccessful connection due to invalid CN in the packet capture. CN as FQDN: • Configure the TOE for reference identifier name as FQDN. • Configure the Server certificate showing invalid CN. • Configure the Server certificate showing no SAN extension. • Establish a TLS connection with the syslog server using above server certificate with acumen-tlsc tool and verify the connection failure. • Verify the connection failure logs on the device. • Verify the unsuccessful connection due to invalid CN in a packet capture.
Expected Test Results	• **The TOE rejects certificates with an invalid CN and No SAN** • **TOE logs should show connection failure due to invalid CN and No SAN** • **Packet capture should show invalid CN and no SAN is configured in the certificate and FIN message is generated by TOE**
Pass/Fail with Explanation	Pass. The TOE rejects certificates with an invalid CN and No SAN.

7.2.6.13 FCS_TLSC_EXT.1.2 Test#2

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall

	repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.
Test Steps	CN and SAN as IPV4: • Configure the TOE for reference identifier name as IPV4. • Configure the Server certificate showing valid CN but invalid SAN. • Initiate the connection from the TOE to the TLS Server and verify that the connection fails. • Verify the connection failure logs on the device. • Verify the unsuccessful connection due to invalid SAN but the CN matches with the reference identifier in a packet capture. CN and SAN as FQDN: • Configure the TOE for reference identifier name as FQDN. • Configure the Server certificate showing valid CN but invalid SAN. • Initiate the connection from the TOE to the TLS Server and verify that the connection fails. • Verify the connection failure logs on the device. • Verify the unsuccessful connection due to invalid SAN but the CN matches with the reference identifier in a packet capture.
Expected Test Results	• **The TOE rejects certificates with a good CN but bad SAN.** • **TOE logs should show connection failure due to SAN mismatch.** • **Packet capture should show valid CN and invalid SAN in configured in the certificate and FIN message is generated by TOE.**
Pass/Fail with Explanation	Pass. The TOE rejects certificates with a good CN but bad SAN.

7.2.6.14 FCS_TLSC_EXT.1.2 Test#3

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.
Test Steps	CN as IPv4 • Configure the TOE with an IPv4 address as reference identifier. • Create a certificate with no SAN and a CN that matches the configured reference identifier. • Initiate a connection from the TOE and verify that it succeeds. • Verify successful connection via packet capture. CN as FQDN

	• Configure the TOE with an FQDN as reference identifier. • Create a certificate with no SAN and a CN that matches the configured reference identifier. • Initiate a connection from the TOE and verify that it succeeds. • Verify successful connection via packet capture.
Expected Results	**When a server certificate contains a CN that matches the reference identifier and does not contains the SAN extension, the TOE accepts the connection.**
Pass/Fail with Explanation	Pass. The TOE does not mandate the presence of the SAN extension. Test #1 was tested only keeping CN and no SAN which proves the same.

7.2.6.15 FCS_TLSC_EXT.1.2 Test#4

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).
Test Steps	CN and SAN as IPV4: • Configure the TOE for reference identifier as IPv4 address. • Configure the Server certificate having invalid CN but a valid SAN extension. • Establish a connection with the TOE over TLS and verify that the connection succeeds. • Verify the successful connection due to SAN matching the reference identifier on the TOE despite an invalid CN in a packet capture. CN and SAN as FQDN: • Configure the TOE for reference identifier name as FQDN. • Configure the Server certificate having invalid CN but a valid SAN extension. • Establish a connection with the TOE over TLS and verify that the connection succeeds. • Verify the successful connection due to SAN matching the reference identifier on the TOE despite an invalid CN in a packet capture.
Expected Test Results	• **The TOE accepts the connection when the certificate with an invalid CN and valid SAN is presented.** • **TOE logs and packet capture should show a successful connection when the certificate with an invalid CN and valid SAN is presented.**
Pass/Fail with Explanation	Pass. The TOE accepts the connection when the certificate with an invalid CN and valid SAN is presented

7.2.6.16 FCS_TLSC_EXT.1.2 Test#5(1)

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** Test 5: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID): The evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails.
Test Steps	For CN: • Configure the TOE for reference identifier name as FQDN. • Create a server certificate with a wildcard that is not present in the left-most label of CN. • Initiate a TLS connection from the TOE to the syslog server using acumen-tlsc tool. • Verify through packet capture that the connection doesn't succeed. • Verify through logs that connection has been terminated. For SAN: • Configure the TOE for reference identifier name as FQDN. • Create a server certificate with a wildcard that is not present in the left-most label of SAN. • Initiate a TLS connection from the TOE to the syslog server using acumen-tlsc tool. • Verify through packet capture that the connection doesn't succeed. • Verify through logs that connection has been terminated.
Expected Test Results	• **TOE rejects the connection when the reference identifier does not match the presented wildcard which is not in the leftmost label.** • **TOE logs should show connection failure.** • **Packet capture should show FIN message is generated by TOE due to mismatched parameters.**
Pass/Fail with Explanation	Pass. The TOE rejects the connection when the reference identifier does not match the presented wildcard which is not in the leftmost label.

7.2.6.17 FCS_TLSC_EXT.1.2 Test#5(2)(a)

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** Test 5: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID): The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com).

	The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds, if wildcards are supported, or fails if wildcards are not supported. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)
Test Steps	Note: The TOE supports wildcards in the SAN extension, but not in the CN. CN: • Configure the TOE for the reference identifier with a single left-most label. • Configure the node certificate showing wildcard in the leftmost label in CN. • Establish a connection with the TOE over TLS and verify the failed connection. • Verify the failed connection logs on the device. • Verify the failed connection via packet capture. SAN: • Configure the TOE for the reference identifier with a single left-most label. • Configure the node certificate showing wildcard in the leftmost label in SAN. • Establish a connection with the TOE over TLS and verify the successful connection. • Verify the successful connection via packet capture.
Expected Test Results	• **TOE rejects the connection when the reference identifier with single left-most labels is presented in the CN of the certificate.** • **TOE logs and packet capture should show a failed connection.** • **TOE accepts the connection when the reference identifier with single left-most labels is presented in the SAN of the certificate.** • **TOE logs and packet capture should show a successful connection**
Pass/Fail with Explanation	Pass. TOE rejects the connection when wildcards are not supported and accepts the same when they are.

7.2.6.18 FCS_TLSC_EXT.1.2 Test#5(2)(b)

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** Test 5: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID): The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails.

	(Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)
Test Steps	Note: The TOE supports wildcards in the SAN extension, but not in the CN. CN: • Configure the TOE for the reference identifier without a leftmost label. • Configure the node certificate showing wildcard in the leftmost label in CN. • Establish a connection with the TOE over TLS and verify that it fails. • Verify the error logs on the device. • Verify the unsuccessful connection with packet capture. SAN: • Configure the TOE for the reference identifier without a leftmost label. • Configure the node certificate showing wildcard in the leftmost label in SAN. • Establish a connection with the TOE over TLS and verify that it fails. • Verify the error logs on the device. • Verify the unsuccessful connection with packet capture.
Expected Test Results	• **When configured with a reference identifier with no left-most labels, the TOE rejects the connection when presented with a server certificate containing a wildcard in the left-most label.** • **TOE logs should show connection failure.** • **Packet capture should show FIN message is generated by TOE due to mismatched parameters.**
Pass/Fail with Explanation	Pass. When configured with a reference identifier with no left-most labels, the TOE rejects the connection when presented with a server certificate containing a wildcard in the left-most label.

7.2.6.19 FCS_TLSC_EXT.1.2 Test#5(2)(c)

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** Test 5: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID): The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)
Test Steps	Note: The TOE supports wildcards in the SAN extension, but not in the CN.

	CN: • Configure the TOE for the reference identifier with two leftmost labels. • Configure the node certificate showing wildcard in the leftmost label in CN. • Establish a connection with the TOE over TLS and verify that the connection fails. • Verify the failure logs on the TOE. • Verify the unsuccessful connection via packet capture. SAN: • Configure the TOE for the reference identifier with two leftmost labels. • Configure the node certificate showing wildcard in the leftmost label in SAN. • Establish a connection with the TOE over TLS and verify that the connection fails. • Verify the failure logs on the TOE. • Verify the unsuccessful connection via packet capture.
Expected Test Results	• **When configured with a reference identifier with two left-most labels, the TOE rejects the connection when presented with a server certificate containing a wildcard in the left-most label.** • **TOE logs should show connection failure.** • **Packet capture should show FIN message is generated by TOE due to mismatched parameters.**
Pass/Fail with Explanation	Pass. When configured with a reference identifier with two left-most labels, the TOE rejects the connection when presented with a server certificate containing a wildcard in the left-most label.

7.2.6.20 FCS_TLSC_EXT.1.2 Test#6

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT**. If IP addresses are supported, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with an asterisk (*) (e.g. CN=192.168.1.* when connecting to 192.168.1.20, CN=2001:0DB8:0000:0000:0008:0800:200C:* when connecting to 2001:0DB8:0000:0000:0008:0800:200C:417A). The certificate shall not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported IP address version (e.g. IPv4, IPv6). Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 6.
Test Steps	• Configure the TOE for the correct IPv4 reference identifier. • Create a server certificate without a SAN extension and with a CN that matches the IPv4 reference identifier but replace one of the groups with an *.

	• Initiate a connection from the TOE over TLS and verify that the connection fails. • Verify the failure logs on the device. • Verify the unsuccessful connection with packet capture.
Expected Test Results	• **TOE rejects the connection when configured server certificate has a missing SAN extension and contains a CN that matches the reference identifier IP with one of the groups replaced with an asterisk (*).** • **TOE generates failure logs.** • **Packet capture showing failure due to CN mismatch.**
Pass/Fail with Explanation	Pass. TOE rejects the connection when configured server certificate has a missing SAN extension and contains a CN that matches the reference identifier IP with one of the groups replaced with an asterisk (*). This meets the test requirements.

7.2.6.21 FCS_TLSC_EXT.1.2 Test#7a

Item	Data
Test Assurance Activity	If **the secure channel is used for FPT_ITT, and RFC 5280 is selected**, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator modifies each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier): The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails.
Pass/Fail with Explanation	NA. RFC 5280 is not selected in the ST.

7.2.6.22 FCS_TLSC_EXT.1.2 Test#7b

Item	Data
Test Assurance Activity	If **the secure channel is used for FPT_ITT, and RFC 5280 is selected**, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator modifies each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier): The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-at-serialNumber=correct_identifier, the certificate could instead include id-at-

	name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass this test.
Pass/Fail with Explanation	NA. RFC 5280 is not selected in the ST.

7.2.6.23 FCS_TLSC_EXT.1.2 Test#7c

Item	Data
Test Assurance Activity	If **the secure channel is used for FPT_ITT, and RFC 5280 is selected**, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator modifies each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier): The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.
Pass/Fail with Explanation	NA. RFC 5280 is not selected in the ST.

7.2.6.24 FCS_TLSC_EXT.1.2 Test#7d

Item	Data
Test Assurance Activity	If **the secure channel is used for FPT_ITT, and RFC 5280 is selected**, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator modifies each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier): The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)
Pass/Fail with Explanation	NA. RFC 5280 is not selected in the ST.

7.2.6.25 FCS_TLSC_EXT.1.3 Test#1

Item	Data
Test Assurance Activity	Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.
Pass/Fail with Explanation	Pass. This testing was performed in conjunction with FIA_X509_EXT.1.1/Rev Test #1a to demonstrate correct operation.

7.2.6.26 FCS_TLSC_EXT.1.3 Test#2

Item	Data
Test Assurance Activity	The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.
Pass/Fail with Explanation	Pass. No administrator override mechanism is supported. Connection failure for invalid server certificates was covered by FIA_X509_EXT.1/Rev and FIA_X509_EXT.2/Rev tests.

7.2.6.27 FCS_TLSC_EXT.1.3 Test#3

Item	Data
Test Assurance Activity	The purpose of this test to verify that only selected certificate validation failures could be administratively overridden. **If any override mechanism is defined for failed certificate validation**, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.
Pass/Fail with Explanation	NA. TOE does not implement any administrator override mechanism as per ST.

7.2.6.28 FCS_TLSC_EXT.1.4 Test#1

Item	Data

Test Assurance Activity	If the TOE presents the **Supported Elliptic Curves/Supported Groups Extension**, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.
Test Steps	• Initiate the connection from the TOE to the TLS Server using acumen-tlsc tool, specifying the curve secp256r1 and verify the successful connection. • Verify with packet capture that the used curve is secp256r1. • Initiate the connection from the TOE to the TLS Server using acumen-tlsc tool, specifying the curve secp384r1. Verify the successful connection. • Verify with packet capture that the used curve is secp384r1.
Expected Test Results	• **The TOE accepts a connection when supported curves were used.** • **Packet capture shows a successful connection.**
Pass/Fail with Explanation	Pass. The TOE accepted a connection when supported curves were used. This meets the test requirements.

7.2.7 TLSS

7.2.7.1 FCS_TLSS_EXT.1.1 Test#1

Item	Data
Test Assurance Activity	Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	• Upload a CA-signed certificate chain to be used for the Manager GUI. **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256** • Establish a TLS connection from the VM with the TOE using the TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ciphersuite. • Verify that the session was established with the chosen ciphersuite. • Verify that the session was established with the chosen ciphersuite via packet capture. **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384** • Establish a TLS connection from the VM with the TOE using the TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA384 ciphersuite. • Verify that the session was established with the chosen ciphersuite. • Verify that the session was established with the chosen ciphersuite via packet capture.
Expected Test Results	• **TOE logs should show a successful establishment of TLS connection.**

	• **Packet captures show the successful establishment of TLS connection with configured ciphersuites.**
Pass/Fail with Explanation	Pass. The TOE was able to make successful connection via the supported ciphersuites. This meets the test requirements.

7.2.7.2 FCS_TLSS_EXT.1.1 Test#2

Item	Data
Test Assurance Activity	Test 2: The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the server denies the connection.
Test Steps	• Attempt a TLS connection to the TOE from the VM with the unsupported TLS_ECDHE_RSA_WITH_AES_128_SHA256 ciphersuite. • Verify that the connection is rejected via packet capture. • Verify the rejected connection via logs. • Attempt a TLS connection to the TOE from the VM with the NULL ciphersuite. • Verify that the connection is rejected via packet capture. • Verify the rejected connection via logs.
Expected Test Results	• **Connection should be rejected when an unsupported ciphersuite or the NULL ciphersuite is used.** • **Packet capture shows handshake failure when using unsupported or NULL ciphersuites**
Pass/Fail with Explanation	Pass. The TOE rejects TLS connections with the unsupported ciphersuites. This meets the testing requirement.

7.2.7.3 FCS_TLSS_EXT.1.1 Test#3a

Item	Data
Test Assurance Activity	Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data.
Test Steps	• Run the acumen-tlss-test tool as a client to modify a byte in the client finished message. The connection should fail. • Verify the unsuccessful connection via packet capture. • Verify the unsuccessful connection via logs.
Expected Test Results	• **TOE should reject a connection when the byte in client finished handshake message is modified.** • **Packet capture and logs should show connection failure when the Client Finished handshake message is modified.** • **The TOE should generate the appropriate logs for failure.**
Pass/Fail with Explanation	Pass. The TOE rejects the connection after receiving the modified Client Finished message. This meets the test requirements.

7.2.7.4 FCS_TLSS_EXT.1.1 Test#3b

Item	Data
Test Assurance Activity	(Test Intent: The intent of this test is to ensure that the server's TLS implementation immediately makes use of the key exchange and authentication algorithms to: a) Correctly encrypt (D)TLS Finished message and b) Encrypt every (D)TLS message after session keys are negotiated.) The evaluator shall use one of the claimed ciphersuites to complete a successful handshake and observe transmission of properly encrypted application data. The evaluator shall verify that no Alert with alert level Fatal (2) messages were sent. The evaluator shall verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator shall examine the Finished message (encrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 11 22 33 44 55...) and confirm that it does not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c...), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. There is a chance that an encrypted Finished message contains a hexadecimal value of '14' at the position where a plaintext Finished message would contain the message type code '14'. If the observed Finished message contains a hexadecimal value of '14' at the position where the plaintext Finished message would contain the message type code, the test shall be repeated three times in total. In case the value of '14' can be observed in all three tests it can be assumed that the Finished message has indeed been sent in plaintext and the test has to be regarded as 'failed'. Otherwise it has to be assumed that the observation of the value '14' has been due to chance and that the Finished message has indeed been sent encrypted. In that latter case the test shall be regarded as 'passed'.
Test Steps	• Initiate a TLS connection to the TOE with the acumen-tlss-test tool as a client. • Verify that no Alert with alert level Fatal (2) messages were sent. • Verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. • Examine the Finished message and confirm that it does not contain unencrypted data by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages.
Expected Test Results	• **TOE should accept an appropriately encrypted TLS connection.** • **Evidence (Packet capture) showing the message is encrypted hence the connection is successful.**
Pass/Fail with Explanation	Pass. The Finished message contains Hexadecimal 16 and is sent immediately after Hexadecimal 14 in the ChangeCipherSpec message. The first byte of the encrypted Finished message does not equal hexadecimal 14. This meets the testing requirement.

7.2.7.5 FCS_TLSS_EXT.1.2 Test#1

Item	Data
Test Assurance Activity	The evaluator shall send a Client Hello requesting a connection for all mandatory and selected protocol versions in the SFR (e.g. by enumeration of protocol versions in a test client) and verify that the server denies the connection for each attempt.
Test Steps	Use the acumen-tlss-test tool as a client to initiate a connection to the TOE and verify the connections fails for all the non-supported SSL and TLS versions. ○ Verify the connection fails with SSLv2.0. ○ Verify failure using packet capture. ○ Verify failure using logs. ○ Verify the connection fails with SSLv3.0. ○ Verify failure using packet capture. ○ Verify failure using logs. ○ Verify the connection fails with TLSv1.0. ○ Verify failure using packet capture. ○ Verify failure using logs. ○ Verify the connection fails with TLSv1.1. ○ Verify failure using packet capture. ○ Verify failure using logs.
Expected Test Results	• **Server should reject a connection when a client requests a connection with the unsupported TLS/SSL versions.** • **TOE logs should show connection failure due to an unknown protocol version.** • **Packet capture should show connection failure due to unsupported protocol version.**
Pass/Fail with Explanation	Pass. The TOE rejects all SSLv2, SSLv3, TLS v1.0 and TLS v1.1 connection attempts. This meets the testing requirement.

7.2.7.6 FCS_TLSS_EXT.1.3 Test#1a

Item	Data
Test Assurance Activity	If **ECDHE ciphersuites** are supported: The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single supported elliptic curve specified in the Elliptic Curves Extension. The Evaluator shall verify (though a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange message and successfully establishes the connection.
Test Steps	• Initiate a TLS connection with the TOE using the secp256r1 curve and verify that the connection is successful

	• Verify the packet capture showing successful connection using the secp256r1 curve.
Expected Test Results	• **The connection should be successful when a supported ECDHE cipher and elliptic curve are configured.** • **Evidence (Packet capture) showing the supported elliptic curve.**
Pass/Fail with Explanation	Pass. The TOE was able to make connection using each supported elliptic curve. This meets the test requirements.

7.2.7.7 FCS_TLSS_EXT.1.3 Test#1b

Item	Data
Test Assurance Activity	If **ECDHE ciphersuites** are supported: The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g. secp192r1 (0x13)) specified in RFC4492, chap. 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.
Test Steps	• Attempt a TLS connection to the TOE using a supported ciphersuite and the unsupported secp192r1 elliptical curve and verify that the connection fails. • Verify the packet capture showing connection failure. • Verify the logs showing connection failure.
Expected Test Results	• **Connection should be rejected when supported cipher and the unsupported elliptic curve are configured.** • **Evidence (Packet capture and logs) showing connection failure with the unsupported elliptic curve.** • **The TOE must generate the appropriate logs indicating failure.**
Pass/Fail with Explanation	Pass. The TOE rejects a connection with unsupported elliptic curves. This meets the testing requirements.

7.2.7.8 FCS_TLSS_EXT.1.3 Test#2

Item	Data
Test Assurance Activity	If **DHE ciphersuites** are supported, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use a supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the message are the ones configured Diffie-Hellman parameter size(s).
Pass/Fail with Explanation	NA. No DHE ciphersuites are supported.

7.2.7.9 FCS_TLSS_EXT.1.3 Test#3

Item	Data
Test Assurance Activity	If **RSA key establishment ciphersuites** are supported, the evaluator shall repeat this test for each RSA key establishment key size. If any configuration is necessary, the evaluator shall configure the TOE to perform RSA key establishment using a supported key size (e.g. by loading a certificate with the appropriate key size). The evaluator shall attempt a connection using a supported RSA key establishment ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a certificate whose modulus is consistent with the configured RSA key size.
Pass/Fail with Explanation	NA. No RSA key establishment ciphersuites are supported.

7.2.7.10 FCS_TLSS_EXT.1.4 Test#1

Item	Data
Test Assurance Activity	If the **TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077**, the evaluator shall perform the following test: a) The client sends a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket. b) The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake). c) The client verifies the Server Hello message contains a zero-length session identifier or passes the following steps: Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID. d) The client completes the TLS handshake and captures the SessionID from the ServerHello. e) The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d). f) The client verifies the TOE: a. implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or b. terminates the connection in some way that prevents the flow of application data. **TD0569 has been applied**
Pass/Fail with Explanation	NA. The TOE supports session resumption.

7.2.7.11 FCS_TLSS_EXT.1.4 Test#2a

Item	Data
Test Assurance Activity	If the **TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2)**, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246). **TD0569 has been applied**
Test Steps	• Use the openssl s_client -sess_out and -sess_in options to save and resume a session using session ID respectively. • Verify via packet capture that the Client Hello uses the previously captured session ID, to which the TOE responds with a Server Hello containing the same session ID, immediately followed by the ChangeCipherSpec and Finished messages.
Expected Test Results	• **TOE accepts a TLS connection that uses a session ID captured from a previously successful and valid TLS session.** • **TOE resumes the previous session by responding with a ServerHello message containing the same SessionID.**
Pass/Fail with Explanation	Pass. The TOE resumed a previously successful and valid TLS session when presented with the captured session ID.

7.2.7.12 FCS_TLSS_EXT.1.4 Test#2b

Item	Data
Test Assurance Activity	If the **TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2)**, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. **TD0569 has been applied**
Test Steps	• Use the acumen-tlss-test tool to: ○ initiate a TLS handshake and disrupt it by generating a fatal alert immediately before the client ChangeCipherSpec message, capturing the session ID in the process.

	○ Initiate a new Client Hello using the previously captured session ID • Verify via packet capture that the TOE implicitly rejects the session ID by sending a ServerHello containing a different SessionID and completes the handshake.
Expected Test Results	• **The server does not resume an invalid session.** • **The server implicitly rejects the previously captured session ID from an invalid session by sending one of its own.**
Pass/Fail with Explanation	Pass. The TOE implicitly rejects the previously used session ID for an invalid session and sends a ServerHello containing a different session ID and completes the handshake. This meets the testing requirement.

7.2.7.13 FCS_TLSS_EXT.1.4 Test#3a

Item	Data
Test Assurance Activity	If the **TOE supports session tickets according to RFC5077**, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with an abbreviated handshake described in section 3.1 of RFC 5077 and illustrated with an example in figure 2. Of particular note: if the server successfully verifies the client's ticket, then it may renew the ticket by including a NewSessionTicket handshake message after the ServerHello in the abbreviated handshake (which is shown in figure 2). This is not required, however as further clarified in section 3.3 of RFC 5077. **TD0569 has been applied.**
Pass/Fail with Explanation	NA. The TOE does not support session resumption based on session tickets.

7.2.7.14 FCS_TLSS_EXT.1.4 Test#3b

Item	Data
Test Assurance Activity	If the **TOE supports session tickets according to RFC5077**, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of

	RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data. **TD0569 has been applied.**
Pass/Fail with Explanation	NA. The TOE does not support session resumption based on session tickets.

7.2.8 TLSS-MA

7.2.8.1 FCS_TLSS_EXT.2.1&2 Test#1a

Item	Data
Test Assurance Activity	If the **TOE requires or can be configured to require a client certificate**, the evaluator shall configure the TOE to require a client certificate and send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify that the handshake is not finished successfully and no application data flows.
Test Steps	• Use the acumen-tlss-test tool to attempt to establish a TLS connection to the TOE while sending a zero length certificate and verify that it fails. • Verify the failure logs on the device. • Verify the failed connection via packet capture.
Expected Test Results	• **TOE rejects the TLS connection when the client does not provide its certificate.** • **Logs show the failed TLS connection.**
Pass/Fail with Explanation	Pass. The TOE rejects the connection when the client tries to connect with the zero-length certificate. This meets the test requirements.

7.2.8.2 FCS_TLSS_EXT.2.1&2 Test#1b

Item	Data
Test Assurance Activity	If the **TOE supports fallback authentication functions and these functions cannot be disabled**. The evaluator shall configure the fallback authentication functions on the TOE and configure the TOE to send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify the TOE authenticates the connection using the fallback authentication functions as described in the TSS. Note: Testing the validity of the client certificate is performed as part of X.509 testing.
Pass/Fail with Explanation	NA. The TOE does not provide any fallback authentication functions.

7.2.8.3 FCS_TLSS_EXT.2.1&2 Test#2

Item	Data
Test Assurance Activity	If **TLS 1.2 is claimed for the TOE**, the evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify that the connection is denied.
Test Steps	• Use the acumen-tlss tool to attempt a TLS connection, modifying the signature algorithm used by the client certificate in the 'Certificate Verify' message to an unsupported one, and verify that the connection fails. • Verify the failure logs on the device. • Verify via packet capture that the handshake fails, and no application data flows.
Expected Test Results	• **The TOE denies a TLS connection inititated using a client certificate without the supported_signature_algorithm.** • **Logs show the failed TLS connection.**
Pass/Fail with Explanation	Pass. The TOE rejects a mutually authenticated TLS connection attempt from a client containing an unsupported signature algorithm.

7.2.8.4 FCS_TLSS_EXT.2.1&2 Test#3

Item	Data
Test Assurance Activity	The aim of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To carry out this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognised by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not terminate in the claimed CA certificate). The evaluator shall verify that the attempted connection is denied.
Test Steps	• Verify the original TOE ICA. • Create an impostor ICA certificate whose DN matches with the original CA certificate on the TOE but with a different key. • Replace the good ICA with the impostor ICA in the CA chain on the sensor. • Verify that the connection and hence the trust establishment between the manager and the sensor fails. • Verify the failure logs on the TOE. • Verify failure via packet capture.
Expected Test Results	• **TOE rejects a TLS connection initiated using an impostor CA.** • **Logs show the connection failure.**

Pass/Fail with Explanation	Pass. The TOE rejects a TLS connection when a certificate signed with the wrong key is presented from a client.

7.2.8.5 FCS_TLSS_EXT.2.1&2 Test#4

Item	Data
Test Assurance Activity	The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that the server denies the connection. Ideally, the two certificates should be identical except for the Client Authentication purpose.
Test Steps	With Client Authentication Extended Key Usage: • Use a sensor certificate containing the Client Authentication purpose. • Verify that the connection and hence the trust establishment between the manager and the sensor succeeds. • Verify via packet capture that the Client Authentication purpose was present. Without Client Authentication Extended Key Usage: • Use a sensor certificate missing the Client Authentication purpose. • Replace the good sensor leaf certificate with the above one. • Verify that the connection and hence the trust establishment between the manager and the sensor fails. • Verify the failure logs on the TOE. • Verify via packet capture that the connection fails.
Expected Test Results	• **TOE accepts a TLS connection initiated using a client certificate containing the Client Authentication purpose.** • **TOE denies a TLS connection initiated using a client certificate missing the Client Authentication purpose.** • **TOE logs show the failed connection.**
Pass/Fail with Explanation	Pass. The TOE denies the connection when the client certificate is missing the Client Authentication purpose. This meets the test requirements.

7.2.8.6 FCS_TLSS_EXT.2.1&2 Test#5a

Item	Data
Test Assurance Activity	Configure the server to require mutual authentication and then connect to the server with a client configured to send a client certificate that is signed by a Certificate Authority trusted by the TOE. The evaluator shall verify that the server accepts the connection.
Test Steps	• Upload the complete sensor certificate chain to the Manager. • Initiate a TLS connection with the TOE and verify that it is successful.

	• Verify successful connection via packet capture.
Expected Test Results	**TOE accepts a TLS connection when presented with a client certificate signed by a CA trusted by the TOE**
Pass/Fail with Explanation	Pass. TOE accepts the connection for a client certificate trusted by TOE. This meets the testing requirements.

7.2.8.7 FCS_TLSS_EXT.2.1&2 Test#5b

Item	Data
Test Assurance Activity	Configure the server to require mutual authentication and then modify a byte in the signature block of the client's Certificate Verify handshake message (see RFC5246 Sec 7.4.8). The evaluator shall verify that the server rejects the connection.
Test Steps	• Use the acumen-tlss-test tool to initiate a TLS connection to the TOE with a modified byte in the signature block of the client certificate and verify that it gets rejected. • Verify failure logs. • Verify failure via packet capture.
Expected Test Results	**TOE rejects a TLS connection when presented with a client certificate with a modified byte in the signature block.**
Pass/Fail with Explanation	Pass. The TOE rejects a TLS connection when it receives a modified signature block in the client certificate. This meets the test requirements.

7.2.8.8 FCS_TLSS_EXT.2.1&2 Test#6

Item	Data
Test Assurance Activity	Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.
Pass/Fail with Explanation	Pass. Requirements are met by FCS_TLSS_EXT.2.1&2 Test #5a.

7.2.8.9 FCS_TLSS_EXT.2.1&2 Test#7

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

	trusted channel was established) covering the types of failure for which an override mechanism is defined.
Pass/Fail with Explanation	Pass. The test requirements are covered by FCS_TLSS_EXT.2.3 Test #1 and the FIA_X509_EXT.1.1/ITT tests.

7.2.8.10 FCS_TLSS_EXT.2.1&2 Test#8

Item	Data
Test Assurance Activity	The purpose of this test is to verify that only selected certificate validation failures could be administratively overridden. **If any override mechanism is defined for failed certificate validation**, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.
Pass/Fail with Explanation	NA. The TOE does not implement any administrator override mechanism.

7.2.8.11 FCS_TLSS_EXT.2.3 Test#1

Item	Data
Test Assurance Activity	The evaluator shall send a client certificate with an identifier that does not match an expected identifier and verify that the server denies the connection.
Test Steps	• Create a client certificate containing an unexpected reference identifier. • Initiate a TLS connection to the TOE with the above client certificate and verify that it fails. • Verify failure logs. • Verify connection failure via packet capture.
Expected Test Results	**TOE denies a TLS connection initiated using a client certificate containing an unexpected reference identifier.**
Pass/Fail with Explanation	Pass. TOE rejects the connection when the certificate with incorrect reference identifier was received from the client.

7.2.9 Update

7.2.9.1 FPT_TST_EXT.1 Test#1

Item	Data
Test Assurance Activity	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. 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. 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.
Test Steps	• Login to the TOE via console and issue the command to reboot. • Observe that the self-tests are completed.
Expected Test Results	• **The TOE should execute all claimed self-tests during bootup.** • **Evidence (screenshot or CLI output) showing successful self-tests**
Pass/Fail with Explanation	Pass. The TOE performs all claimed self-tests successfully. This meets the testing requirements.

7.2.9.2 FPT_TUD_EXT.1 Test#1

Item	Data
Test Assurance Activity	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.
Test Steps	• Show the current version. • Upgrade the TOE using a valid image. • Show the new version post upgrade and reboot. • Verify the successful upgrade via logs.
Expected Test Results	• **The TOE should successfully update the current version with the new version after verifying the integrity of the new image.** • **Evidence - screenshot showing new version post upgrade.**

Pass/Fail with Explanation	Pass. The TOE successfully updates to a new version using a valid image.

7.2.9.3 FPT_TUD_EXT.1 Test#2(a)

Item	Data
Test Assurance Activity	Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below, and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates: 1) A modified version (e.g. using a hex editor) of a legitimately signed update 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.
Test Steps	• Verify the current version of the TOE. • Using a Hex editor modify an otherwise good firmware image. • Issue the 'upgrade' command on the manager which will import the corrupted image from an update server. Verify the manager upgrade fails while the corrupted image is being applied to the manager during the verification process. • Verify via logs that the upgrade fails. • Verify that the TOE version remains unchanged.
Expected Test Results	• **The TOE should detect and reject the modified image for software update.** • **Evidence – TOE logs showing software upgrade failure logs.**
Pass/Fail with Explanation	Pass. The TOE detects and rejects the modified image for software update

7.2.9.4 FPT_TUD_EXT.1 Test#2(b)

Item	Data

Test Assurance Activity	[conditional]: If **the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE** the following test shall be performed (otherwise the test shall be omitted). The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below, and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates: 2) An image that has not been signed 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.
Test Steps	• Verify the current version of the TOE. • Upload an image file that has not been signed and verify that the update fails. • Verify the failed software upgrade via logs. • Verify that the TOE version remains unchanged.
Expected Test Results	• **The TOE should detect and reject the image without signature for a software update.** • **Software upgrade failed logs generated on TOE**
Pass/Fail with Explanation	Pass. The TOE detects and reject the image without signature for a software update.

7.2.9.5 FPT_TUD_EXT.1 Test#2(c)

Item	Data
Test Assurance Activity	[conditional]: If **the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE** the following test shall be performed (otherwise the test shall be omitted). The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below, and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates: 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) 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.
Test Steps	• Verify the current image version on the TOE. • Attempt an update using an image with an invalid signature and verify that it fails. • Verify the logs reflecting failure. • Verify the TOE image was not upgraded.
Expected Test Results	• **The TOE should detect and reject the image with an invalid signature for the software update.** • **Software upgrade failed logs generated on TOE.**
Pass/Fail with Explanation	Pass. The TOE detects and rejects the image with an invalid signature for the software update.

7.2.9.6 FPT_TUD_EXT.1 Test #2 (d)

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	NA. The TOE does not support delayed activation of updates.

7.2.9.7 FPT_TUD_EXT.1 Test #3 (a)

Item	Data
Test Assurance Activity	[conditional]: If the TOE itself verifies a hash value over an image against a published hash value (i.e. reference value) that has been imported to the TOE from outside such that the TOE itself authorizes the installation of an image to update the TOE, the following test shall be performed (otherwise the test shall be omitted. If the published hash is provided to the TOE by the Security Administrator and the verification of the hash value over the update file(s) against the published hash is performed by the TOE, then the evaluator shall perform the following tests. The evaluator first confirms that no update is pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test: 1) The evaluator obtains or produces an illegitimate update such that the hash of the update does not match the published hash. The evaluator provides the published hash value to the TOE and calculates the hash of the update either on the TOE itself (if that functionality is provided by the TOE), or else outside the TOE. The evaluator confirms that the hash values are different, and attempts to install the update on the TOE, verifying that this fails because of the difference in hash values (and that the failure is logged). Depending on the implementation of the TOE, the TOE might not allow the Security Administrator to even attempt updating the TOE after the

	verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE. 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.
Pass/Fail with Explanation	NA. The TOE does not support published hash verification.

7.2.9.8 FPT_TUD_EXT.1 Test #3 (b)

Item	Data
Test Assurance Activity	[conditional]: If **the TOE itself verifies a hash value over an image against a published hash value (i.e. reference value) that has been imported to the TOE from outside such that the TOE itself authorizes the installation of an image to update the TOE,** the following test shall be performed (otherwise the test shall be omitted. If the published hash is provided to the TOE by the Security Administrator and the verification of the hash value over the update file(s) against the published hash is performed by the TOE, then the evaluator shall perform the following tests. The evaluator first confirms that no update is pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test: 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. 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.
Pass/Fail with Explanation	NA. The TOE does not support published hash verification.

7.2.9.9 FPT_TUD_EXT.1 Test #3 (c)

Item	Data
Test Assurance Activity	[conditional]: If **the TOE itself verifies a hash value over an image against a published hash value (i.e. reference value) that has been imported to the TOE from outside such that the TOE itself authorizes the installation of an image to update the TOE,** the following test shall be performed (otherwise the test shall be omitted. If the published hash is provided to the TOE by the Security Administrator and the verification of the hash value over the update file(s) against the published hash is performed by the TOE, then the evaluator shall perform the following tests. The evaluator first confirms that no update is pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test: 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.
Pass/Fail with Explanation	NA. The TOE does not support delayed activation of updates.

7.2.10 X509-ITT

7.2.10.1 FIA_X509_EXT.1.1/ITT Test#1a

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function, and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate

	needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).
Test Steps	• Generate a CSR on the TOE. • Export and sign the CSR. • Load the Server Certificate chain on the TOE. • Verify logs confirming the successful certificate import onto the TOE. • Configure the TOE to communicate with the Sensor using full certificate chain. • Initiate a connection to the TOE and verify that it is successful. • Verify the successful connection via PCAP.
Expected Test Results	**The TOE allows a connection when a proper chain is present.**
Pass/Fail with Explanation	Pass. The TOE successfully connects to the sensor when a complete certificate chain is exchanged. This meets the testing requirements.

7.2.10.2 FIA_X509_EXT.1.1/ITT Test#1b

Item	Data
Test Assurance Activity	Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.
Test Steps	• Verify the CA certificate chain located on the sensor. • Remove the ICA from the CA certificate chain. • Verify that the sensor's trust with NSM breaks leading to connection failure. • Verify the connection failure logs on the TOE. • Verify connection failure via packet capture.
Expected Test Results	**When a complete chain of certificates is not presented, the TOE doesn't establish a successful TLS connection. The packet capture depicts that FIN packets were sent to indicate that the connection has been terminated.**
Pass/Fail with Explanation	Pass. When a complete chain of certificates is not presented, the TOE doesn't establish a successful TLS connection.

7.2.10.3 FIA_X509_EXT.1.1/ITT Test#2

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that

	utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.
Test Steps	• Verify the time on TOE. • Create an expired sensor certificate. • Verify the sensor's certificate. • Replace the sensor's certificate with the expired certificate. • Verify that the connection and hence the trust establishment fails. • Verify the failure logs on the TOE. • Verify connection failure via packet capture.
Expected Test Results	**When an expired certificate is presented, the TOE terminates the TLS connection. The packet capture shows that the connection was terminated and FIN packets were sent to indicate that connection was not established successfully.**
Pass/Fail with Explanation	Pass. When an expired certificate is presented, the TOE terminates the TLS connection.

7.2.10.4 FIA_X509_EXT.1.1/ITT Test#3

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. Test 3: The evaluator shall test that the TOE can properly handle revoked certificates—conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the TOE certificate and revocation of the TOE intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. No testing is required if no revocation method is selected. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.
Pass/Fail with Explanation	N/A. No revocation checking is specified for this component.

7.2.10.5 FIA_X509_EXT.1.1/ITT Test#4

Item	Data

Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.
Pass/Fail with Explanation	N/A. No revocation checking is specified for this component.

7.2.10.6 FIA_X509_EXT.1.1/ITT Test#5

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)
Test Steps	• Use the acumen-tlss tool to modify the first eight bytes of the certificate and attempt a connection. This should fail. • Verify that the connection is terminated using logs. • Verify the connection failure via packet capture.
Expected Test Results	**The TOE denies a TLS connection when it is presented with a certificate that has been modified using the 'acumen-tlss' tool. The tool modifies the first eight bytes of the certificate. The packet capture verifies that the connection is not established due to the bad certificate.**
Pass/Fail with Explanation	Pass. The TOE rejects a connection when the first 8 bytes of the certificate are modified. This meets the testing requirements.

7.2.10.7 FIA_X509_EXT.1.1/ITT Test#6

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
Test Steps	• Verify the sensor's certificate. • Modify the last byte in the sensor certificate. • Verify that the connection and hence the trust establishment fails. • Verify the failure logs on the TOE. • Verify connection failure via packet capture.

Expected Test Results	• **The TOE fails to establish a TLS connection when the last byte in the signatureValue field of the certificate is modified. The packet capture proves that there is a decrypt error and the logs show that there is a failure in establishing connection due to certificate signature failure.**
Pass/Fail with Explanation	Pass. The TOE fails to establish a TLS connection when the last byte in the signatureValue field of the certificate is modified.

7.2.10.8 FIA_X509_EXT.1.1/ITT Test#7

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.)
Test Steps	• Use the acumen-tlss tool to modify a byte of the public key of the certificate and try to establish a connection using this certificate. Verify the connection fails. • Verify that the connection is terminated using logs. • Verify the connection failure via packet capture.
Expected Test Results	**The TOE rejects a remote TLS connection that is formed using the 'acumen-tlss' tool. The tool modifies the certificate such that its public key is modified and uses the same certificate for establishing the TLS connection. The packet capture depicts that there TCP FIN packets showing connection termination due to certificate signature failure.**
Pass/Fail with Explanation	Pass. The TOE rejects a connection when the public key of the certificate is modified. This meets the testing requirements.

7.2.10.9 FIA_X509_EXT.1.1/ITT Test#8a

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** **(Conditional on TOE ability to process CA certificates presented in certificate message)** The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain. **TD0527 (12/1 Update) has been applied.**

Pass/Fail with Explanation	N/A. EC signatures are not supported for the TLS functionality.

7.2.10.10 FIA_X509_EXT.1.1/ITT Test#8b

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** **(Conditional on TOE ability to process CA certificates presented in certificate message)** The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid. **TD0527 (12/1 Update) has been applied.**
Pass/Fail with Explanation	N/A. EC signatures are not supported for the TLS functionality.

7.2.10.11 FIA_X509_EXT.1.1/ITT Test#8c

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store. **TD0527 (12/1 Update) has been applied.**
Pass/Fail with Explanation	N/A. EC signatures are not supported for the TLS functionality.

7.2.10.12 FIA_X509_EXT.1.2/ITT Test#1

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for FIA_X509_EXT.1.2/ITT. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/ITT. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted. The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation). For each of the following tests the evaluator shall create a chain of at least two certificates: a self-signed root CA certificate a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain). Test 1: The evaluator shall ensure that one CA in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).
Test Steps	• Create an ICA Certificate with no basicConstraints extension. • Replace the good ICA in the sensor certificate chain with the above one. • Attempt to load the modified certificate chain to the TOE and verify that it gets rejected. • Verify that the certificates are rejected using logs.
Expected Test Results	**The TOE rejects a certificate where the CA certificate doesn't contain the basicConstraints extension. The logs depict the rejection.**
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that does not contain the basicConstraints extension. This meets the testing requirements.

7.2.10.13 FIA_X509_EXT.1.2/ITT Test#2

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for FIA_X509_EXT.1.2/ITT. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/ITT. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE

	(i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted. The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation). For each of the following tests the evaluator shall create a chain of at least two certificates: a self-signed root CA certificate a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain). Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).
Test Steps	• Create an ICA Certificate with the CA flag in the basicConstraints extension set to FALSE. • Replace the good ICA in the sensor certificate chain with the above one. • Attempt to load the modified certificate chain to the TOE and verify that it gets rejected. • Verify that the certificates are rejected using logs.
Expected Test Results	**The TOE rejects a certificate where the CA certificate contains the CA flag in the basicConstraints extension set to FALSE. The logs depict the rejection.**
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that has the CA flag in the basicConstraints extension set to FALSE. This meets the testing requirements.

7.2.10.14 FIA_X509_EXT.2 Test#1

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each trusted channel: The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the guidance documentation to determine that all supported administrator-configurable options behave in their documented manner.
Pass/Fail with Explanation	N/A. No revocation checking is specified for this component.

7.2.10.15 FIA_X509_EXT.3 Test#1

Item	Data
Test Assurance Activity	Test 1: The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated request and ensure that it conforms to the format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.
Test Steps	• Generate a CSR on the TOE and export it. • Examine the CSR contents and verify that it contains the following subject fields: Common Name, Organization, Organizational Unit, Country. Also verify that it contains details about the public key and other details.
Expected Test Results	**The TOE is able to generate a CSR with the required fields**
Pass/Fail with Explanation	Pass. The TOE is able to generate a CSR with all of the requisite information. This meets the testing requirements.

7.2.10.16 FIA_X509_EXT.3 Test#2

Item	Data
Test Assurance Activity	Test 2: The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the response message, and demonstrate that the function succeeds."
Test Steps	• Generate a CSR on the TOE. • Export and sign the generated CSR using the ICA. • Remove the ICA certificate from the TOE certificate chain. • Attempt to load the above invalid certificate chain on the TOE. • Verify that the TOE rejects the certificate chain because the full trust chain is not present. • Verify that logs confirm the failure in loading of TOE certificates. • Add back the ICA to the TOE certificate chain to ensure that it now has a full certificate path. • Re-attempt to load the now valid certificate chain on the TOE. • Verify that the TOE accepts the certificate because the path validation succeeded. • Verify that logs confirm the successful loading of TOE certificates.
Expected Test Results	**The TOE will reject signed certificates if there is no trust chain or if the chain is broken; the TOE will accept certificates for full valid chains.**
Pass/Fail with Explanation	Pass. The TOE does not install CSR responses signed by a CA without a full trust path. This meets the testing requirements.

7.2.11 X509-Rev

7.2.11.1 FIA_X509_EXT.1.1/Rev Test#1a

Item	Data
Test Assurance Activity	Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).
Test Steps	• Upload a complete certificate chain to the TOE. • Configure the TOE to connect to the syslog server. • Initiate a connection from the TOE to the syslog server over TLS and verify the successful connection. • Verify TOE logs for successful connection. • Verify the successful connection with packet capture.
Expected Test Results	• **When a complete certificate chain is present, the TOE should establish a successful TLS connection.** • **TOE logs and packet capture should show a successful connection as a complete chain of certificates is present on the TOE.**
Pass/Fail with Explanation	Pass. The TOE can make a successful connection when a complete certificate trust chain is present. This meets the test requirements.

7.2.11.2 FIA_X509_EXT.1.1/Rev Test#1b

Item	Data
Test Assurance Activity	Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.
Test Steps	• Remove the ICA from the syslog server CA chain. • Initiate a connection from the TOE to the syslog server over TLS and verify that the connection fails. • Verify the connection failure via logs on the TOE. • Verify the unsuccessful connection with packet capture.

Expected Test Results	• **The TOE rejects the connection when an incomplete certificate trust chain is present.** • **TOE log should show failure due to the untrusted certificate being used.** • **Packet capture showing connection failure as intermediate CA certificate is removed from TOE.**
Pass/Fail with Explanation	Pass. The TOE rejects the connection when an incomplete certificate trust chain is present. This meets the test requirements.

7.2.11.3 FIA_X509_EXT.1.1/Rev Test#2

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.
Test Steps	• Create a syslog server certificate that is expired. • Show the clock on the TOE. • Initiate a connection from the TOE to the syslog server using an expired server certificate and verify that it fails. • Verify the failure logs on the device, showing connection is not established due to expired certificate. • Verify the connection is unsuccessful via packet capture.
Expected Test Results	• The TOE should deny connection when the certificate is expired. • TOE logs showing connection failure due to an expired server certificate. • Packet capture showing connection failure as expired server certificate is used.
Pass/Fail with Explanation	Pass. A connection using an expired certificate was rejected. This meets the test requirements.

7.2.11.4 FIA_X509_EXT.1.1/Rev Test#3

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE.

	Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.
Test Steps	Valid Certificate: • Create a server certificate containing the OCSP responder URL. • Create root CA with OCSP Signing and ICA certificate with OCSP URL. • Create another CA certificate signed by the ICA, to be delegated with OCSP signing for the server certificate. • Generate an index file and setup OCSP responders. • Attempt a connection and verify that it is successful. • Verify the successful connection and OCSP responses with packet capture. Revoked End Entity Certificate: • Revoke the server certificate. • Generate an index file and setup OCSP responders. • Attempt a connection with the TOE and verify that it fails. • Verify the failure logs on the TOE showing validation failed due to revoked certificate. • Verify the unsuccessful connection and OCSP response with packet capture. Invalid Intermediate CA Certificate: • Revoke the ICA certificate. • Generate an index file and setup OCSP responders. • Attempt a connection with the TOE and verify that it fails. • Verify the failure logs on the TOE showing validation failed due to revoked certificate. • Verify the unsuccessful connection and OCSP response with packet capture.
Expected Test Results	• **The TOE rejects any TLS server connection when either the intermediate certificate or the server certificate has been revoked.** • **The OCSP connection also shows that the certificates have been revoked.** • **The Packet capture depicts the specific certificate that has been revoked and the logs verify that the TOE has denied connection by denoting that certificate has been revoked.**
Pass/Fail with Explanation	Pass. The TOE rejects connections that use revoked certificates. This meets the testing requirements.

7.2.11.5 FIA_X509_EXT.1.1/Rev Test#4

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.
Test Steps	• Generate an OCSP signer certificate that does NOT have the OCSP signing EKU. • Set up an OCSP responder for the server certificate using this signer certificate. • Attempt a connection from TOE to the syslog server using OpenSSL and verify that it fails. • Verify the unsuccessful TLS connection with the help of packet capture. • Verify validation of certificate is failed as the signer certificate doesn't have OCSP signing EKU via TOE logs.
Expected Test Results	• **The TOE doesn't establish a TLS server connection when the OCSP signing purpose is missing and validation fails.** • **The packet capture shows that there is a handshake failure due to the absence of OCSP Signing.** • **The logs are used to validate the fact that the connection has been rejected by OCSP due to a failure in certificate verification.**
Pass/Fail with Explanation	Pass. The TOE rejects connections when the Signer certificate in OCSP is invalid and does not have signing purpose. This meets the testing requirements.

7.2.11.6 FIA_X509_EXT.1.1/Rev Test#5

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)
Test Steps	• Configure the TOE to connect to the syslog server. • Use the acumen-tlsc tool to modify a byte within the first 8 bytes of the certificate, the attempted connection should fail. • Verify the error logs on the TOE showing failure. • Verify the unsuccessful connection with packet capture.
Expected Test Results	• **TOE rejects connections when the first 8 bytes of the certificate are modified.** • **TOE should generate error logs when a certificate with modified bytes is presented.**

	• **Packet capture showing connection failure due to certificate with modified bytes is presented.**
Pass/Fail with Explanation	Pass. TOE rejects connections when the first 8 bytes of the certificate are modified. This meets the test requirements.

7.2.11.7 FIA_X509_EXT.1.1/Rev Test#6

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. The evaluator shall modify any byte in the certificate signatureValue field (see RFC5280 Sec. 4.1.1.3), which is normally the last field in the certificate, and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
Test Steps	• Configure the TOE to connect to the syslog server. • Start the server using the acumen-tlsc tool to modify a byte in the signatureValue field of the certificate. • Verify the error with logs on the device showing certificate verification failed. • Verify the unsuccessful connection with packet capture.
Expected Test Results	• **TOE rejects connections when the last byte of the certificate is modified.** • **TOE should generate error logs when a certificate with modified bytes is presented.** • **Packet capture showing connection failure due to certificate with modified bytes is presented.**
Pass/Fail with Explanation	Pass. The TOE rejects connections when the byte in the certificate signatureValue field is modified. This meets the test requirements.

7.2.11.8 FIA_X509_EXT.1.1/Rev Test#7

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.)
Test Steps	• Configure the TOE to connect to the syslog server. • Start the server using the acumen-tlsc tool to modify a byte in the public key of the certificate. • Verify the error logs on the device showing failure due to an invalid public key.

	• Verify the unsuccessful connection with packet capture.
Expected Test Results	• **The TOE rejects connections when the public key of the certificate is modified** • **TOE should generate error logs showing failure due to an invalid public key** • **Packet capture should show connection failure as the certificate with the modified public key is presented**
Pass/Fail with Explanation	Pass. The TOE rejects connections when any byte is the public key of the certificate is modified. This meets the test requirements.

7.2.11.9 FIA_X509_EXT.1.1/Rev Test#8a

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** **(Conditional on TOE ability to process CA certificates presented in certificate message)** The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain. **TD0527 (12/1 Update) has been applied.**
Pass/Fail with Explanation	NA. The TOE does not support ECDSA signatures for the TLS functionality.

7.2.11.10 FIA_X509_EXT.1.1/Rev Test#8b

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** **(Conditional on TOE ability to process CA certificates presented in certificate message)** The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid. **TD0527 (12/1 Update) has been applied.**

Pass/Fail with Explanation	NA. The TOE does not support ECDSA signatures for the TLS functionality.

7.2.11.11 FIA_X509_EXT.1.1/Rev Test#8c

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store. **TD0527 (12/1 Update) has been applied.**
Pass/Fail with Explanation	NA. The TOE does not support ECDSA signatures for the TLS functionality.

7.2.11.12 FIA_X509_EXT.1.2/Rev Test#1

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted. The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation). For each of the following tests the evaluator shall create a chain of at least three certificates: - a self-signed root CA certificate, - an intermediate CA certificate and - a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain). Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: *(i) as part of the validation of the leaf certificate belonging to this chain;*

	(ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).
Test Steps	• Configure the TOE to connect to the syslog server. • Create an ICA with no basicConstraint field. • Replace the good ICA with the above one in the CA certificate chain. • Use the modified certificate chain to start the syslog server, attempt to establish a connection and verify that it gets rejected by the TOE. • Verify the error logs on the device showing the certificate getting rejected due to missing basic constraints. • Verify the connection failure via packet capture.
Expected Test Results	• **The TOE should reject certificates signed by CA that do not contain the BasicConstraints Extension** • **Toe should generate error logs showing the certificate rejected due to basic constraint failure**
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that do not contain the basicConstraints extension. This meets the test requirements.

7.2.11.13 FIA_X509_EXT.1.2/Rev Test#2

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted. The goal of the following tests it to verify that the TOE accepts only certificates that have been marked as CA certificates by using basicConstraints with the CA flag set to True (and implicitly that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation). For each of the following tests the evaluator shall create a chain of at least three certificates: - a self-signed root CA certificate, - an intermediate CA certificate and - a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain). Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: 1. As part of the validation of the leaf certificate belonging to this chain; When attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

Test Steps	• Create an ICA with basicConstraint field set to FALSE. • Replace the good ICA with the above one in the CA certificate chain. • Use the modified certificate chain to start the syslog server, attempt to establish a connection and verify that it gets rejected by the TOE. • Verify the error logs on the device showing the certificate getting rejected due to missing basic constraints. • Verify the connection failure via packet capture.
Expected Test Results	• **The TOE should reject certificates signed by CA that has the CA flag set to FALSE.** • **TOE should generate error logs showing the certificate rejected due to basic constraint failure.**
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that has the CA flag in the basicConstraints extension set to FALSE. This meets the test requirements.

7.2.11.14 FIA_X509_EXT.2 Test#1

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each trusted channel: The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the guidance documentation to determine that all supported administrator-configurable options behave in their documented manner.
Test Steps	• Create a server certificate with a modified URL. • Start the OCSP Responder along with the syslog server. • Attempt a connection and verify that it fails • Verify the error in logs. • Verify the connection failure via packet capture.
Expected Test Results	• **The TOE will reject the OCSP connection as the certificate used has an incorrect URL.** • **The packet capture will depict a handshake failure while the logs should show a failure in establishing a connection.**
Pass/Fail with Explanation	Pass. The TOE rejects certificates it cannot verify via OCSP when the responder is down. This meets the testing requirements.

7.2.11.15 FIA_X509_EXT.3 Test#1

Item	Data
Test Assurance Activity	The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated message and ensure that it conforms to the

	format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.
Pass/Fail with Explanation	NA. The connection with the syslog server does not have mutual authentication and since the TOE behaves as a client, it does not need a certificate, thereby not supporting a CSR generation functionality. To be tested under X509-ITT.

7.2.11.16 FIA_X509_EXT.3 Test#1

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message and demonstrate that the function succeeds.
Pass/Fail with Explanation	NA. The connection with the syslog server does not have mutual authentication and since the TOE behaves as a client, it does not need a certificate, thereby not supporting a CSR generation functionality. To be tested under X509-ITT.

8 Conclusion

The testing shows that all test cases required for conformance have passed testing.

End of Document