Trellix Intrusion Prevention System Sensor and Manager Appliances version 11.1 Update 7 Security Target

Document Version: 2.0

intertek
acumen
security

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History

Version	Date	Changes
Version 1.0	Dec 28, 2022	Initial Release
Version 1.1	Jul 16, 2023	Updated based on Key Destruction table.
Version 1.2	Sep 01, 2023	Updated formatting for Check-in.
Version 1.3	Oct 16, 2023	Updated for ECR comments.
Version 1.4	Nov 15, 2023	Updated for 2nd Round of ECR comments.
Version 1.5	Feb 23, 2024	Updated to include additional Sensor devices and changes to FCS_SSHC_EXT.1.5 and FMT_SMF.1.1 selections.
Version 1.6	March 18, 2024	Updated TSS section to add additional information for multiple SFRs along with changes to FCS_COP.1/Hash and FCS_COP.1/KeyedHash selections.
Version 1.7	April 15, 2024	Minor updates based on internal reviews
Version 1.8	May 08, 2024	Updated as per the ECR comments
Version 1.9	May 14, 2024	Updated as per the 2nd round of ECR comments
Version 2.0	November 11, 2024	Updated for Assurance Maintenance

Contents

1 Introduction

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE, and the functional and assurance requirements that the TOE meets.

1.1 Security Target and TOE Reference

This section provides the information needed to identify and control the TOE and the ST.

Table 1 – TOE/ST Identification

Category	Identifier
ST Title	Trellix Intrusion Prevention System Sensor and Manager Appliances version 11.1 Update 7 Security Target
ST Version	2.0
ST Date	November 11, 2024
ST Author	Acumen Security, LLC.
TOE Identifier	Trellix Intrusion Prevention System Sensor and Manager Appliances version 11.1 Update 7
TOE Version	Sensor version 11.1.17.46 and Manager version 11.1.19.46
TOE Developer	Trellix (Musarubra US LLC)
Key Words	Network Device, Intrusion Prevention System (IPS)

1.2 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. These models are listed in Table 2.

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.

1.2.1 IPS Manager Architecture

The Manager Appliance is the 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 and push configuration data and policies to the Sensors.

1.2.2 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:

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

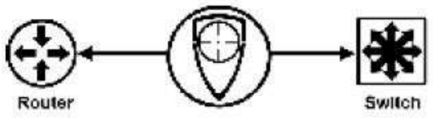

Figure 1: Sensor in "Inline" mode

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.

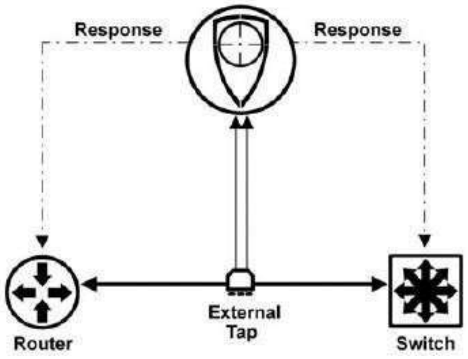

Figure 2: Sensor in "Tap" mode

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.

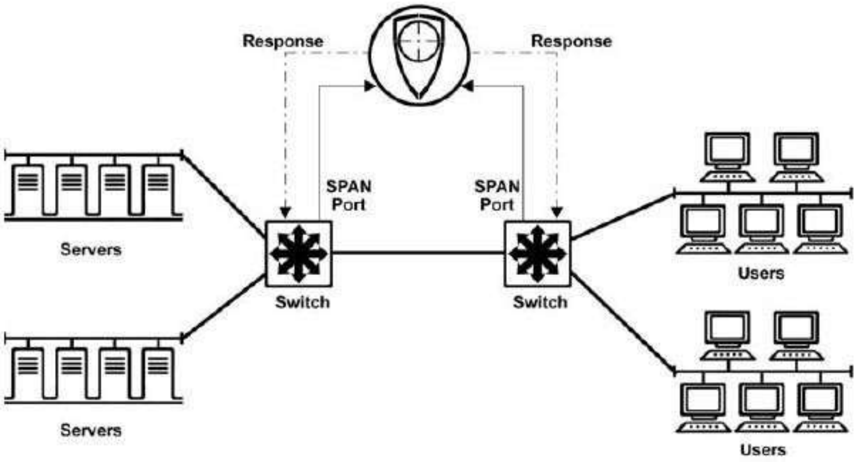

Figure 3: Sensor in "Span" mode

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.

1.3 TOE Description

The TOE is a distributed TOE. This section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references.

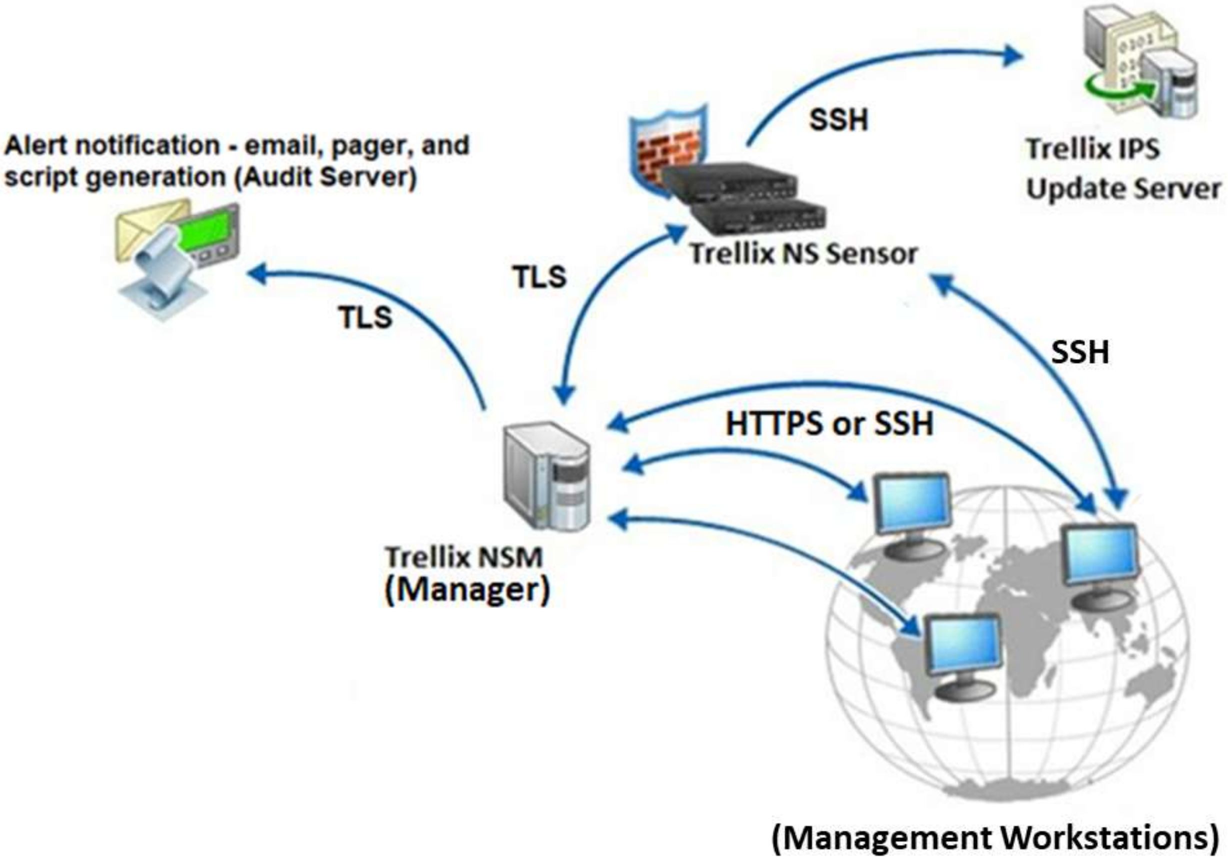
Alert notification - email, pager, and script generation (Audit Server)
TLS
TLS
SSH
Trellix NS Sensor
Trellix IPS Update Server
SSH
HTTPS or SSH
Trellix NSM (Manager)
(Management Workstations)

Figure 4 – Representative TOE Deployment

1.3.1 Physical Boundaries

The TOE is distributed. It is a combination of:

- One or more IPS Sensor appliances with their software [Sensor]
- One IPS Manager appliance with its software [Manager]

Each component is delivered with the TOE software installed. The following table lists all the instances of the Sensors that are included in the evaluation. All listed Sensor appliances offer the same security functionality but vary in the type and number of processors, amount of memory, and storage.

Table 2 - TOE Appliance Series and Models

Model	CPUs	Memory (Size and Qty)	Storage	Micro-architecture
Trellix Intrusion Prevention System Sensor Appliances				
IPS-NS9600	2 x XEON GOLD 5520+	12 x 32 GB	2 x 400GB SSD	Emerald Rapids
IPS-NS9500	2 x XEON GOLD 6230	12 x 16GB	2 x 240GB SSD	Cascade Lake
IPS-NS7600	1 x XEON SILVER 4416+	6 x 32GB	1 x 400GB SSD	Sapphire Rapids
IPS-NS7500	1 x XEON GOLD 5218N	6 x 16GB	1 x 240GB SSD	Cascade Lake

Model	CPUs	Memory (Size and Qty)	Storage	Micro-architecture
IPS-NS3600	1 x XEON D-1734NT	2 x 32GB	1 x 400GB SSD	Ice Lake
IPS-NS3200	1 x ATOM C2538	2 x 4GB	1 x 30GB SSD	Rangeley
Trellix Intrusion Prevention System Manager Appliance				
NSM-MAPL-NG	1 x XEON SILVER 4210	4 x 16GB	2 x 2TB HDD	Cascade Lake
NSM-MAPL -NG	1 x XEON SILVER 4114	4 x 16GB	2 x 2TB HDD	Skylake

In the evaluated configuration, the devices are placed in Network Device collaborative Protection Profile (NDcPP) mode by configuration according to the Administrative Guidance.

1.3.2 Security Functions Provided by the TOE

The TOE provides the security functions required by the Collaborative Protection Profile for Network Devices, hereafter referred to as NDcPP v2.2e or NDcPP.

1.3.2.1 Security Audit

The TOE generates audit records related to TOE operation and administration. These audit records are stored on the IPS Manager (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 Sensor generates audit records and forwards the audit records to the IPS Manager, the Sensor caches audit records in a local file. The audit file can be uploaded to Manager (or any other SCP server using the “auditlogupoload” CLI command).

Only authenticated users can view audit records.

1.3.2.2 Communication

The TOE is a Distributed TOE. It is a combination of:

- One or more IPS Sensor appliances with their software [Sensor]
- One IPS Manager appliance with its software [Manager]

Each component is delivered with the TOE software installed. A security Administrator can enable or disable communications between any pair of TOE components. The communication between the TOE components is secured via TLS with Mutual Authentication as per the secure channel requirements in FPT_ITT.1.

1.3.2.3 Cryptographic Support

The TOE uses symmetric key cryptography to secure communication between the Sensors and the Manager for the following functionality:

- Exchange of configuration information (including IPS policies)
- Time/date synchronization from the Manager to Sensors
- Transfer of IPS data to the Manager
- Transfer of audit records to the Manager
- Distribution of TOE updates to Sensors

Connections between the Manager and Sensors are secured using TLS.

Connections between the Manager and the Audit Server (for audit record upload) are secured using TLS.
Connection between a Sensor and the Update Server is secured using SSH.
Sessions between the Management Workstation and the TOE are secured using SSH or HTTPS. Administrators can connect to the Manager via HTTPS or SSH. Administrators can connect to the Sensor via SSH.
Local console connections between the Console Workstation and the TOE are physically secured.

For all cryptographic operations performed by the TOE, the cryptographic algorithms have been validated as identified in the table below.

Table 3 – Manager CAVP Certificate References

The following table presents a listing of each IPS Manager algorithm certificates and the associated OE.
[NSM-MAPL-NG (XEON SILVER 4210) and NSM-MAPL -NG (XEON SILVER 4114)]

Functions	Algorithms	Mode Supported	IPS CAVP Certs.	Name	OE
Data Encryption	AES-GCM	GCM (128, 256)	A4660	Network Security Manager Bouncy Castle	MLOS 3 on Intel Xeon Scalable Processors (Silver 4114, Skylake) MLOS 3 on Intel Xeon Scalable Processors (Silver 4210, Cascade Lake)
			A2624	Trellix OpenSSL FIPS Object Module	
Hash	SHS (Cryptographic hashing)	SHA-1, SHA-256, SHA-384, SHA-512	A4660	Network Security Manager Bouncy Castle	
			A2624	Trellix OpenSSL FIPS Object Module	
Random Number Generation	Counter DRBG	CTR_DRBG (AES-256)	A4660	Network Security Manager Bouncy Castle	
			A2624	Trellix OpenSSL FIPS Object Module	
Key Generation	RSA KeyGen (FIPS186-4)	Mode: n(2048), n = 2048 SHA(256)	A4660	Network Security Manager Bouncy Castle	
			A2624	Trellix OpenSSL FIPS Object Module	
	ECDSA KeyGen (FIPS186-4)	P-256, P-384	A4660	Network Security Manager Bouncy Castle	

	ECDSA KeyVer (FIPS186-4)		A2624	Trellix OpenSSL FIPS Object Module	
Key Establishment	KAS ECC Sp800-56Ar3 (Key Pair Generation, Partial Validation)	P-256, P-384	A4660	Network Security Manager Bouncy Castle	
Digital Signature services	ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)	P-256	A4660	Network Security Manager Bouncy Castle	
			A2624	Trellix OpenSSL FIPS Object Module	
	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	Mode: n(2048), n = 2048 SHA(256)	A4660	Network Security Manager Bouncy Castle	
			A2624	Trellix OpenSSL FIPS Object Module	
Keyed Hash	HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	Mode: SHA-256, SHA-384, SHA-512	A4660	Network Security Manager Bouncy Castle	
			A2624	Trellix OpenSSL FIPS Object Module	

The following table presents a listing of each IPS Sensor algorithm certificates and the associated OEs. [NS9600, NS9500, NS7600, NS7500, NS3600, NS3200]

Table 4 - Sensor CAVP Certificate References

Functions	Algorithms	Mode Supported	IPS CAVP Certs.	Name	OE

Data Encryption	AES-GCM	GCM (128, 256)	A3350	Trellix IPS Sensor Crypto Lib	Intel(R) Atom(R)C Series (C2538, Rangeley)
					Intel(R) Xeon(R) (D-1734NT, Ice Lake)
					Intel(R) Xeon(R) Scalable Processors (GOLD 5218N, Cascade Lake)
					Intel(R) Xeon(R) Scalable Processors (GOLD 6230, Cascade Lake)
					Intel(R) Xeon(R) Silver (4416+, Sapphire Rapids)

Hash	SHS (Cryptographic hashing)	SHA-1, SHA-256, SHA-384, SHA-512	A3350	Trellix IPS Sensor Crypto Lib	
		SHA-256, SHA-384	A3353	Trellix IPS Sensor XySSL Lib	
Random Number Generator	Counter DRBG	CTR_DRBG (AES-256)	A3350	Trellix IPS Sensor Crypto Lib	
Key Generation	RSA KeyGen (FIPS186-4)	Mode: n(2048), n = 2048 SHA(256)	A3350	Trellix IPS Sensor Crypto Lib	
	ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4)	P-256, P-384	A3350	Trellix IPS Sensor Crypto Lib	
Key Establishment	KAS ECC SSC Sp800-56Ar3 (Domain Parameter Generation)	P-256, P-384	A3350	Trellix IPS Sensor Crypto Lib	

Digital Signature services	ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)	P-256	A3350	Trellix IPS Sensor Crypto Lib	Intel(R) Xeon(R) Gold (5520+, Emerald Rapids)
	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	Mode: n(2048), n = 2048 SHA(256)	A3350	Trellix IPS Sensor Crypto Lib	
	RSA SigVer (FIPS186-4)		A3353	Trellix IPS Sensor XySSL Lib	
Keyed Hash	HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	Mode: SHA-256, SHA-384, SHA-512	A3350	Trellix IPS Sensor Crypto Lib	

1.3.2.4 Identification and Authentication

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

The Manager and Sensors authenticate to one another through a shared secret that is configured during the initial installation and setup process of the TOE. Although in the evaluated configuration, the Manager supports use of a default self-signed certificate for trust establishment with the sensor, such a channel is out of scope for this evaluation. The sensor-Manager channel must be established using CA-signed certificates.

1.3.2.5 Security Management

An administrative CLI can be accessed via the Console port or SSH connection, and an administrative GUI can 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.

1.3.2.6 Protection of the TSF

The presence of the Sensors' components on the network is transparent (other than network packets sent as reactions to be configured IPS conditions). The 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 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 Sensor using the "show" command, which returns details about the software and hardware version. A 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.

1.3.2.7 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.

1.3.2.8 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 for updates

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

1.3.2.9 Intrusion Prevention

The IPS Sensors provides the following IPS-based Functionality:

- Anomaly-based traffic patterns definition, including the specification of frequency and specific network protocol fields
- IP blocking based on known-good and known-bad list of rules, IP addresses (source, destination), ACLs, and alert filters
- IP-based network traffic analysis
- Signature-based traffic analysis

1.3.3 TOE Documentation

The following documents are essential to understanding and controlling the TOE in the evaluated configuration:
https://docs.trellix.com/bundle/ips-landing-page

- Trellix Intrusion Prevention System 11.1.x FIPS and CC Certification Guide – 11.1 Update 7 (November 28, 2024)
- Trellix Intrusion Prevention System 11.1.x Installation Guide – 11.1 Update 7 (November 28, 2024)

- Trellix Intrusion Prevention System Manager Appliance Product Guide – 11.1 Update 7 (November 28, 2024)
- Trellix Intrusion Prevention System NS-series Sensor Product Guide – 11.1 Update 7 (November 28, 2024)
- Trellix Intrusion Prevention System 11.1.x Product Guide – 11.1 Update 7 (November 28, 2024)
- Trellix Intrusion Prevention System NS3600 Sensor Hardware Guide – 11.1 Update 7 (November 28, 2024)
- Trellix Intrusion Prevention System NS7600 Sensor Hardware Guide – 11.1 Update 7 (November 28, 2024)
- Trellix Intrusion Prevention System NS9600 Sensor Hardware Guide – 11.1 Update 7 (November 28, 2024)

1.3.4 References

In addition to TOE documentation, the following references may also be valuable when understanding and controlling the TOE:

- Collaborative Protection Profile for Network Devices, Version 2.2e (CPP_ND_V2.2E)
- PP-Module for Intrusion Protection Systems (IPS), Version 1.0 (MOD_IPS_V1.0)

1.4 TOE Environment

The following environmental components are required to operate the TOE in the evaluated configuration:

Table 5 – Required Environmental Components

Components	Description
Local Management Console	Any computer using terminal emulation software to access the console interface of CLI of the Manager or Sensor.
Remote Management Workstation	Any computer that provides a supported browser may be used to access the Manager via the GUI or using SSH client software to access the CLI.
External IT systems	IT systems exchanging network traffic generate the packets that are analyzed by the TOE.
Update Server	An SCP server used for updating the Sensor software securely over a remoteconnection.
Syslog Server	A syslog server that constantly receives audit logs from the Manager component over a secure TLSv1.2 channel.

1.5 Product Functionality not Included in the Scope of the Evaluation

The following product functionality is not included in the CC evaluation:

- IPS can be configured to maintain accurate time via NTP. NTP must be disabled in the evaluated configuration.
- The Manager can manage Sensors that are not FIPS compliant. All Sensors must be in FIPS mode in the evaluated configuration.
- The Manager can manage Sensors that are using self-signed X.509 certificates. In the evaluated configurations, all Sensors must use CA-signed certificates.

- IPS can be configured to authenticate users via an LDAP server (rather than relying solely on internal user accounts). This optional functionality was not evaluated.

2 Conformance Claims

This section identifies the TOE conformance claims, conformance rationale, and relevant Technical Decisions (TDs).

2.1 CC Conformance Claims

The TOE is conformant to the following:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017 (Extended)
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017 (Conformant)

2.2 Protection Profile Conformance

This ST claims exact conformance to the following:

- **(NDcPP + IPS MOD)** PP-Configuration for Network Device and Intrusion Prevention Systems (IPS), Version 1.0, 18 May 2021

  This PP-Configuration includes the following:
  - collaborative Protection Profile for Network Devices, Version 2.2e (CPP_ND_V2.2E)
  - PP-Module for Intrusion Protection Systems (IPS), Version 1.0 (MOD_IPS_V1.0)

2.3 Conformance Rationale

This ST provides exact conformance to the items listed in the previous section. The security problem definition, security objectives, and security requirements in this ST are all taken from the Protection Profile (PP) and IPS Module, performing only the operations defined there.

2.3.1 Technical Decisions

All NIAP TDs issued to date and applicable to [NDcPP] and [MOD_IPS] have been considered. Table identifies all applicable TDs.

Table 6 – Relevant Technical Decisions

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0527: Updated to Certificate Revocation Testing (FIA_X509_EXT.1)	No	EC signatures are only used for the SSH functionality and not the TLS functionality, which leverages RSA-based X.509v3 certificates.
TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No	This TD addressed NTP functionality, and this TOE does not include NTP functionality.
TD0536: NIT Technical Decision for Update Verification Inconsistency	Yes	
TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Yes	

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0546: NIT Technical Decision for DTLS – clarification of Application Note 63	No	This TD addressed DTLS functionality, and this TOE does not include DTLS functionality.
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	Session resumption using session tickets is not selected.
TD0556: NIT Technical Decisions for RFC 5077 question	No	Session resumption using session tickets is not selected.
TD0563: NIT Technical Decision for Clarification of audit date information	Yes	
TD0564: NIT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	Yes	
TD0570: NIT Technical Decision for Clarification about FIA_AFL.1	Yes	
TD0571: NIT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
TD0572: NIT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
TD0591: NIT Technical Decision for Virtual TOEs and hypervisors	No	Not a Virtual TOE.
TD0592: NIT Technical Decision for Local Storage of Audit Records	Yes	
TD0595: Administrative corrections to IPS PP-Module	Yes	
TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes	
TD0632: NIT Technical Decision for Consistency with Time Data for vNDs	No	TOE does not receive time updates from an underlying virtual server

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters	Yes	
TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH	Yes	
TD0638: NIT Technical Decision for Key Pair Generation for Authentication	Yes	
TD0639: NIT Technical Decision for Clarification for NTP MAC Keys	No	This TD addressed NTP functionality, and this TOE does not include NTP functionality.
TD0670: NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	Yes	
TD0722: IPS_SBD_EXT.1.1 EA Correction	Yes	
TD0738: NIT Technical Decision for Link to Allowed-With List	Yes	
TD0790: NIT Technical Decision: Clarification Required for testing IPv6	No	TOE does not claim IPv6 addresses in CN/SAN
TD0792: NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes	
TD0800: Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	No	TOE does not claim IPsec as a secure channel.

3 Security Problem Definition

The security problem definition has been taken directly from the [NDcPP] and [MOD_IPS] specified in Section 2.2 and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any Organizational Security Policies (OSPs) that the TOE is expected to enforce.

3.1 Threats

The threats included in Table are drawn directly from the [NDcPP] and [MOD_IPS] specified in Section 2.2.

Table 7 – Threats

ID	Threat
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.

ID	Threat
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic and may allow them to take advantage of any trust relationships with other Network Devices.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.
T.NETWORK_DISCLOSURE/IPS	Sensitive information on a protected network might be disclosed resulting from ingress-or egress-based actions.
T.NETWORK_ACCESS/IPS	Unauthorized access may be achieved to services on a protected network from outside that network, or alternately services outside a protected network from inside the protected network. If malicious external devices are able to communicate with devices on the protected network via a backdoor then those devices may be susceptible to the unauthorized disclosure of information.
T.NETWORK_MISUSE/IPS	Access to services made available by a protected network might be used counter to Operational Environment policies. Devices located outside the protected network may attempt to conduct inappropriate activities while communicating with allowed public services. E.g. manipulation of resident tools, SQL injection, phishing,

ID	Threat
	forced resets, malicious zip files, disguised executables, privilege escalation tools, and botnets.
T.NETWORK_DOS/IPS	Attacks against services inside a protected network, or indirectly by virtue of access to malicious agents from within a protected network, might lead to denial of services otherwise available within a protected network. Resource exhaustion may occur in the event of co-ordinate service request flooding from a small number of sources.

3.2 Assumptions

The assumptions included in Table are drawn directly from [NDcPP] and [MOD_IPS].

Table 8 – Assumptions

ID	Assumption
A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).
A.NO_THRU_TRAFFIC_PROTECTION	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).

ID	Assumption
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. (The paragraph that follows is for x509v3 cert-based authentication. If not relevant, remove) For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.COMPONENTS_RUNNING (applies to distributed TOEs only)	For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.
A.CONNECTIONS/IPS	It is assumed that the TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

3.3 Organizational Security Policies

The OSPs included in Table are drawn directly from the [NDcPP] and [MOD_IPS].

Table 9 – OSPs

ID	OSP
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.
P.ANALYZE/IPS	Analytical processes and information to derive conclusions about potential intrusions must be applied to IPS data and appropriate response actions taken.

4 Security Objectives

The security objectives have been taken directly from the [NDcPP] and [MOD_IPS] and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the TOE

The security objectives in the following table apply to the TOE.

Table 10 – Security Objectives

ID	Security Objectives
O.SYSTEM_MONITORING/IPS	The IPS must collect and store information about all events that may indicate an IPS policy violation related to misuse, inappropriate access, or malicious activity on monitored networks.
O.IPS_ANALYZE/IPS	The IPS must apply analytical processes to network traffic data collected from monitored networks and derive conclusions about potential intrusions or network traffic policy violations.
O.IPS_REACT/IPS	The IPS must respond appropriately to its analytical conclusions about IPS policy violations.
O.TOE_ADMINISTRATION/IPS	The IPS will provide a method for authorized administrator to configure the TSF.

4.2 Security Objectives for the Operational Environment

Security objectives for the operational environment assist the TOE in correctly providing its security functionality. These objectives, which are found in the table below, track with the assumptions about the TOE operational environment.

Table 11 – Security Objectives for the Operational Environment

ID	Objectives for the Operational Environment
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration, and support of the TOE. Note: For vNDs the TOE includes only the contents of its own VM, and does not include other VMs or the VS.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

ID	Objectives for the Operational Environment
OE.TRUSTED_ADMN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.COMPONENTS_RUNNING (applies to distributed TOEs only)	For distributed TOEs, the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.
OE.CONNECTIONS/IPS	TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic of monitored networks.

5 Security Requirements

This section identifies the Security Functional Requirements (SFRs) for the TOE. The SFRs included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revisions 5, April 2017, and all international interpretations.

Table 12– SFRs

Requirement	Description	Manager	Sensor
FAU_GEN.1	Audit data generation	Y	Y
FAU_GEN.1/IPS	Audit data generation (IPS)	N	Y
FAU_GEN.2	User identity association	Y	Y
FAU_GEN_EXT.1	Security Audit Data Generation for Distributed TOE component	Y	Y
FAU_STG_EXT.1	Protected audit event storage	Y	N
FAU_STG_EXT.4	Protected Local Audit Event Storage for Distributed TOEs	Y	N
FAU_STG_EXT.5	Protected Remote Audit Event Storage for Distributed TOEs	N	Y
FCO_CPC_EXT.1	Component Registration Channel Definition	Y	Y
FCS_CKM.1	Cryptographic key generation	Y	Y
FCS_CKM.2	Cryptographic key establishment	Y	Y
FCS_CKM.4	Cryptographic key Destruction	Y	Y
FCS_COP.1/ DataEncryption	Cryptographic operation (AES Data Encryption/Decryption)	Y	Y
FCS_COP.1/SigGen	Cryptographic operation (Signature Generation and Verification)	Y	Y
FCS_COP.1/Hash	Cryptographic operation (Hash algorithm)	Y	Y
FCS_COP.1/KeyedHash	Cryptographic operation (Keyed Hash Algorithm)	Y	Y
FCS_HTTPS_EXT.1	HTTPS protocol	Y	N
FCS_RBG_EXT.1	Random bit generation	Y	Y
FCS_SSHC_EXT.1	SSH Client Protocol	N	Y
FCS_SSHS_EXT.1	SSH Server Protocol	Y	Y
FCS_TLSC_EXT.1	TLS client protocol	Y	N
FCS_TLSC_EXT.2	TLS client protocol with authentication	N	Y
FCS_TLSS_EXT.1	TLS server protocol	Y	N
FCS_TLSS_EXT.2	TLS server protocol with authentication	Y	N
FIA_AFL.1	Authentication Failure Management	Y	Y
FIA_PMG_EXT.1	Password management	Y	Y
FIA_UIA_EXT.1	User identification and authentication	Y	Y
FIA_UAU_EXT.2	Password-based authentication mechanism	Y	Y
FIA_UAU.7	Protected authentication feedback	Y	Y
FIA_X509_EXT.1/ITT	X.509 certificate validation	Y	Y

Requirement	Description	Manager	Sensor
FIA_X509_EXT.1/Rev	X.509 certificate validation	Y	N
FIA_X509_EXT.2	X.509 certificate authentication	Y	N
FIA_X509_EXT.3	X.509 certificate requests	Y	Y
FMT_MOF.1/ ManualUpdate	Management of security functions behaviour	Y	Y
FMT_MTD.1/CoreData	Management of TSF data	Y	Y
FMT_SMF.1	Specification of management functions	Y	Y
FMT_SMF.1/IPS	Specification of management functions (IPS)	Y	Y
FMT_SMR.2	Restrictions on security roles	Y	Y
FPT_APW_EXT.1	Protection of administrator passwords	Y	Y
FPT_ITT.1	Basic Internal TSF Data Transfer Protection	Y	Y
FPT_SKP_EXT.1	Protection of TSF data (for reading of all pre-shared, symmetric, and private keys)	Y	Y
FPT_STM_EXT.1	Reliable time stamps	Y	Y
FPT_TST_EXT.1	TSF testing	Y	Y
FPT_TUD_EXT.1	Trusted update	Y	Y
FTA_SSL_EXT.1	TSF-initiated session locking	Y	Y
FTA_SSL.3	TSF-initiated termination	Y	Y
FTA_SSL.4	User-initiated termination	Y	Y
FTA_TAB.1	Default TOE access banners	Y	Y
FTP_ITC.1	Inter-TSF trusted channel	Y	Y
FTP_TRP.1/Admin	Trusted path	Y	Y
IPS_ABD_EXT.1	Anomaly-based IPS functionality	N	Y
IPS_IPB_EXT.1	IP blocking	N	Y
IPS_NTA_EXT.1	Network traffic analysis	N	Y
IPS_SBD_EXT.1	Signature-based IPS functionality	N	Y

5.1 Conventions

The CC allows the following types of operations to be performed on the functional requirements: assignments, selections, refinements, and iterations. The following font conventions are used within this document to identify operations defined by CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration identifier after a slash, e.g., /SigGen.
- Where operations were completed in the PP and relevant EPs/Modules/Packages, the formatting used in the PP has been retained.
- Extended SFRs are identified by the addition of “EXT” after the requirement name.

5.2 Security Functional Requirements

This section includes the security functional requirements for this ST.

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1

The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;

b) Auditable events for the not specified level of audit; and

c) *All administrative actions comprising:*

- *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
- *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
- *Resetting passwords (name of related user account shall be logged).*
- *[no other actions, [[list of other uses of privileges]]];*

d) *Specifically defined auditable events listed in* **Table 13**.

FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of* Table 13*.*

Application Note: This SFR has been updated as per TD0563

Table 13 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.1/IPS	None.	None.
FAU_GEN_EXT.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FAU_STG_EXT.4	None.	None.
FAU_STG_EXT.5	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None	None
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_SSHC_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSC_EXT.2	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.2	Failure to establish a TLS Session	Reason for failure
FCO_CPC_EXT.1	Enabling communications between a pair of components. Disabling communications between a pair of components.	Identities of the endpoints pairs enabled or disabled.
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.1/ITT	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FPT_ITT.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FPT_SKP_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TST_EXT.1	None.	
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FTA_SSL_EXT.1	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the claimed user identity.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the claimed user identity.

5.2.1.2 FAU_GEN.1/IPS Audit Data Generation (IPS)

FAU_GEN.1.1/IPS **Refinement**: The TSF shall be able to generate an **IPS** audit record of the following **IPS** auditable events:

a) Start-up and shut-down of the **IPS** functions;

b) All **IPS** auditable events for the [not specified] level of audit; and

c) [*All dissimilar IPS events;*

d) *All dissimilar IPS reactions;*

e) *Totals of similar events occurring within a specified time period;*

f) *Totals of similar reactions occurring within a specified time period*.

g) The events in the IPS Events table.

h) [no other auditable events]]

Application Note: This SFR has been updated as per TD0595

FAU_GEN.1.2/IPS **Refinement:** The TSF shall record within each **IPS auditable event** record at least the following information:

a) Date and time of the event, type of event **and/or reaction,** ~~subject identity, and the outcome (success or failure) of the event;~~ and;

b) For each **IPS** audit**able** event type, based on the auditable event definitions of the functional components included in the PP~~ST~~, [*information specified in column three of the IPS Events table*].

Table 14 - IPS Events

Requirement	Auditable Events	Additional Audit Record Contents
FMT_SMF.1/IPS	Modification of an IPS policy element.	Identifier or name of the modified IPS policy element (e.g. which signature, baseline, or known-good/known-bad list was modified).
IPS_ABD_EXT.1	Inspected traffic matches an anomaly-based IPS policy.	Source and destination IP addresses. The content of the header fields that were determined to match the policy. TOE interface that received the packet. Aspect of the anomaly-based IPS policy rule that triggered the event (e.g. throughput, time of day, frequency, etc.). Network-based action by the TOE (e.g. allowed, blocked, sent reset to source IP, sent blocking notification to firewall).
IPS_IPB_EXT.1	Inspected traffic matches a list of known-good or known-bad addresses applied to an IPS policy.	Source and destination IP addresses (and, if applicable, indication of whether the source and/or destination address matched the list). TOE interface that received the packet. Network-based action by the TOE (e.g. allowed, blocked, sent reset).

IPS_NTA_EXT.1	Modification of which IPS policies are active on a TOE interface. Enabling/disabling a TOE interface with IPS policies applied. Modification of which mode(s) is/are active on a TOE interface.	Identification of the TOE interface. The IPS policy and interface mode (if applicable).
IPS_SBD_EXT.1	Inspected traffic matches a signature-based IPS rule with logging enabled.	Name or identifier of the matched signature. Source and destination IP addresses. The content of the header fields that were determined to match the signature. TOE interface that received the packet. Network-based action by the TOE (e.g. allowed, blocked, sent reset).

5.2.1.3 FAU_GEN.2 User Identity Association

FAU_GEN.2.1
For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.1.4 FAU_GEN_EXT.1 Security Audit Data Generation for Distributed TOE Component

FAU_GEN_EXT.1.1
The TSF shall be able to generate audit records for each TOE component. The audit records generated by the TSF of each TOE component shall include the subset of security relevant audit events which can occur on the TOE component.

5.2.1.5 FAU_STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1.1
The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2
The TSF Shall be able to store generated audit data on the TOE itself. In addition [

- *The TOE shall be a distributed TOE that stores audit data on the following TOE components: [Manager],*
- *The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [Sensor]*

].

FAU_STG_EXT.1.3
The TSF shall [*overwrite previous audit records according to the following rule: [oldest log entry is overwritten]*] when the local storage space for audit data is full.

5.2.1.6 FAU_STG_EXT.4 Protected Local Audit Event Storage for Distributed TOEs

FAU_STG_EXT.4.1
The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: *[Manager:* [*overwrite previous audit records according to the following rule*: *[oldest log entry is overwritten]*]*]*.

5.2.1.7 FAU_STG_EXT.5 Protected Remote Audit Event Storage for Distributed TOEs

FAU_STG_EXT.5.1
Each TOE component which does not store security audit data locally shall be able to buffer security audit data locally until it has been transferred to another TOE component that stores or forwards it. All transfer of audit records between TOE components shall use a protected channel according to [*FPT_ITT.1*].

5.2.2 Communication (FCO)

5.2.2.1 FCO_CPC_EXT.1 Component Registration Channel Definition

FCO_CPC_EXT.1.1
The TSF shall require a Security Administrator to enable communications between any pair of TOE components before such communication can take place.

FCO_CPC_EXT.1.2
The TSF shall implement a registration process in which components establish and use a communications channel that uses [

- *A channel that meets the secure channel requirements in [FPT_ITT.1],*

for at least *TSF data*.

FCO_CPC_EXT.1.3
The TSF shall enable a Security Administrator to disable communications between any pair of TOE components.

5.2.3 Cryptographic Support (FCS)

5.2.3.1 FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1
The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*
- *ECC schemes using "NIST curves" [P-256, P-384] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;*

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].~~

5.2.3.2 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1
The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

- *Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";*

] ~~that meets the following: [assignment: list of standards].~~

Application Note: This SFR has been updated as per TD0580 and TD0581

5.2.3.3 FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1
The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- *For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite consisting of zeroes];*
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [*
  - *logically addresses the storage location of the key and performs a* [*single overwrite consisting of* [*zeroes*]];

that meets the following: *No Standard*

5.2.3.4 FCS_COP.1/DataEncryption Cryptographic Operations (AES Data Encryption/Decryption)

FCS_COP.1.1/DataEncryption
The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* [*GCM*] *mode* and cryptographic key sizes [*128 bits, 256 bits*] that meet the following: *AES as specified in ISO 18033-3,* [*GCM as specified in ISO 19772*].

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

FCS_COP.1.1/SigGen
The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- *RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits]*

]
that meet the following: [

- *For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,*
- *For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256]; ISO/IEC 14888-3, Section 6.4*

].

5.2.3.6 FCS_COP.1/Hash Cryptographic Operations (Hash Algorithm)

FCS_COP.1.1/Hash
The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [*SHA-1, SHA-256, SHA-384, SHA-512*] ~~and cryptographic key sizes [*assignment: cryptographic key sizes*~~] and **message digest sizes [*160, 256, 384, 512*] bits** that meet the following: *ISO/IEC 10118-3:2004*.

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

FCS_COP.1.1/KeyedHash
The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [*HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, implicit*] and cryptographic key sizes *[256 bits, 384 bits, 512-bits]* **and message digest sizes [*256, 384, 512*] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.2.3.8 FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1.1
The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2
The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3
If a peer certificate is presented, the TSF shall [not require client authentication] if the peer certificate is deemed invalid.

5.2.3.9 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1
The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [*CTR_DRBG (AES-256)*].

FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[1] platform-based noise source*] with a minimum of [*256 bits*] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

5.2.3.10 FCS_SSHC_EXT.1 SSH Client Protocol

FCS_SSHC_EXT.1.1
The TSF shall implement the SSH protocol in accordance with: RFCs *4251, 4252, 4253, 4254*, [5647, 5656, 8308 section 3.1, 8332].

FCS_SSHC_EXT.1.2
The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, [password-based].

Application Note: This SFR has been updated as per TD0636

FCS_SSHC_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than *[256K]* bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.4
The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [aes128-gcm@openssh.com, *aes256-gcm@openssh.com*].

FCS_SSHC_EXT.1.5
The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa, rsa-sha2-256, rsa-sha2-512, *ecdsa-sha2-nistp256*] as its public key algorithm(s) and rejects all other public key algorithms.

Application Note: This SFR has been updated as per TD0636

FCS_SSHC_EXT.1.6
The TSF shall ensure that the SSH transport implementation uses [*implicit*] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7
The TSF shall ensure that [*ecdh-sha2-nistp256*] and [*no other methods*] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.8
The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

FCS_SSHC_EXT.1.9
The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key and [*no other methods*] as described in RFC 4251 section 4.1.

5.2.3.11 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1
The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [5647, 5656].

FCS_SSHS_EXT.1.2
The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, [password-based].

Application Note: This SFR has been updated as per TD0631

FCS_SSHS_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than *[256K]* bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4
The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [*aes128-gcm@openssh.com, aes256-gcm@openssh.com*].

FCS_SSHS_EXT.1.5
The TSF shall ensure that the SSH public-key based authentication implementation uses [*ecdsa-sha2-nistp256*] as its public key algorithm(s) and rejects all other public key algorithms.
Application Note: This SFR has been updated as per TD0631

FCS_SSHS_EXT.1.6
The TSF shall ensure that the SSH transport implementation uses [*implicit*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7
The TSF shall ensure that [*ecdh-sha2-nistp256*] and [*no other methods*] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8
The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

5.2.3.12 FCS_TLSC_EXT.1 TLS Client Protocol without Mutual Authentication

FCS_TLSC_EXT.1.1
The TSF shall implement [*TLS 1.2 (RFC 5246)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

[

- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC5289*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC5289*

] and no other ciphersuites.

FCS_TLSC_EXT.1.2
The TSF shall verify that the presented identifier matches [*the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, and no other attribute types*].

FCS_TLSC_EXT.1.3
When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [*Not implement any administrator override mechanism*].

FCS_TLSC_EXT.1.4
The TSF shall [*present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: secp256r1, secp384r1 and no other curves/groups*] in the Client Hello.

5.2.3.13 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

Application Note: This SFR applies to connections between the Manager and Sensors.

FCS_TLSC_EXT.2.1
The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

5.2.3.14 FCS_TLSS_EXT.1 TLS Sever Protocol Without Mutual Authentication

FCS_TLSS_EXT.1.1
The TSF shall implement [*TLS 1.2 (RFC 5246)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

[

- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC5289*

- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*

] and no other ciphersuites.

FCS_TLSS_EXT.1.2
The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [*TLS 1.1*].

FCS_TLSS_EXT.1.3
The TSF shall perform key establishment for TLS using [*ECDHE curves [secp256r1] and no other curves*]].

FCS_TLSS_EXT.1.4
The TSF shall support [*session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2)*].

Application Note: This SFR has been updated as per TD0569

5.2.3.15 FCS_TLSS_EXT.2 TLS Sever Support for Mutual Authentication

FCS_TLSS_EXT.2.1
The TSF shall support TLS communication with mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.2
When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [

- *Not implement any administrator override mechanism*

].

FCS_TLSS_EXT.2.3
The TSF shall not establish a trusted channel if the identifier contained in a certificate does not match an expected identifier for the client. If the identifier is a Fully Qualified Domain Name (FQDN), then the TSF shall match the identifiers according to RFC 6125, otherwise the TSF shall parse the identifier from the certificate and match the identifier against the expected identifier of the client as described in the TSS.

5.2.4 Identification and Authentication (FIA)

5.2.4.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within *[3-10]* unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been met, the TSF shall *[prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until [manual account unlocking] is taken by an Administrator; prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed]*.

5.2.4.2 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1
The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [_(“~”, “`”, “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “_”, “+”, “-“, “=”, “[“, “]”, “{“, “}”, “\”, “|”, “;”, “:” “"”, “'”, “,”, “.”, “<”, “>”, “?” and “/”]

b) Minimum password length shall be configurable to between [*8*] and [*64*] characters.

5.2.4.3 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1

The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [*no other actions*].

FIA_UIA_EXT.1.2

The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

5.2.4.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1

The TSF shall provide a local [*password-based*] authentication mechanism to perform local administrative user authentication.

5.2.4.5 FIA_UAU.7.1 Protected Authentication Feedback

FIA_UAU.7.1

The TSF shall provide only *obscured feedback* to the administrative user while the authentication is in progress **at the local console**.

5.2.4.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

FIA_X509_EXT.1.1/Rev

Application Note: This SFR applies to certificate validation performed on connections with external (non-TOE) entities.

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates.**
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [*the Online Certificate Status Protocol (OCSP) as specified in RFC 6960*].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
  - *Server certificates presented for TLS shall have the Server Authentication purpose(id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsagefield.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose(id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

FIA_X509_EXT.1.2/Rev
The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.2.4.7 FIA_X509_EXT.1/ITT X.509 Certificate Validation

FIA_X509_EXT.1.1/ITT
Application Note: This SFR applies to certificate validation performed on connections between Manager and Sensors.
The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation supporting a **minimum path length of two certificates**.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set toTRUE.
- The TSF shall validate the revocation status of the certificate using [*no revocation method*]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Server certificates presented for TLS shall have the Server Authentication purpose(id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose(id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsagefield*.

FIA_X509_EXT.1.2/ITT
The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.2.4.8 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [*TLS*] and [*no additional uses*].

FIA_X509_EXT.2.2
When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [*not accept the certificate*].

Application Note: This SFR has been updated as per TD0537.

5.2.4.9 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1
The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [*Common Name, Organization, Organizational Unit, Country*].

FIA_X509_EXT.3.2
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.2.5 Security Management (FMT)

5.2.5.1 FMT_MOF.1/ManualUpdate Management of Security Functions Behavior

FMT_MOF.1.1/ManualUpdate
The TSF shall restrict the ability to enable the function *to perform manual updates to Security Administrators*.

5.2.5.2 FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1.1/CoreData
The TSF shall restrict the ability to manage the *TSF data to Security Administrators.*

5.2.5.3 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1
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 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;*].

Application Note: This SFR has been updated as per TD0631

5.2.5.4 FMT_SMF.1/IPS Specification of Management Functions (IPS)

FMT_SMF.1.1/IPS The TSF shall be capable of performing the following management functions: [

- *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*]

5.2.5.5 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1
The TSF shall maintain the roles:

- *Security Administrator*

FMT_SMR.2.2
The TSF shall be able to associate users with roles.

FMT_SMR.2.3
The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely;*

are satisfied.

5.2.6 Protection of the TSF (FPT)

5.2.6.1 FTP_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1
The TSF shall store administrative passwords in non-plaintext form.

FPT_APW_EXT.1.2
The TSF shall prevent the reading of plaintext administrative passwords.

5.2.6.2 FPT_ITT.1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1
The TSF shall protect TSF data from disclosure and **detect its** modification when it is transmitted between separate parts of the TOE **through the use of [*TLS*]**.

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

FPT_SKP_EXT.1.1
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.2.6.4 FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1.1
The TSF shall be able to provide reliable time stamps for its own use.

FPT_STM_EXT.1.2
The TSF shall [*allow the Security Administrator to set the time*].

5.2.6.5 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [*during initial start-up (on power on), periodically during normal operation, at the conditions [during installation of a trusted update]*] to demonstrate the correct operation of the TSF: [*cryptographic algorithm known answer tests, entropy health tests, software integrity tests*].

5.2.6.6 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1
The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [no other TOE firmware/software version].

FPT_TUD_EXT.1.2
The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [*no other update mechanism*].

FPT_TUD_EXT.1.3
The TSF shall provide means to authenticate firmware/software updates to the TOE using a [*digital signature*] prior to installing those updates.

5.2.7 TOE Access (FTA)

5.2.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1
The TSF Shall, for local interactive sessions, [

- *terminate the session*]

after a Security Administrator-specified time period of inactivity

5.2.7.2 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1
The TSF shall terminate **a remote** interactive session after a *Security Administrator-configurable time interval of session inactivity.*

5.2.7.3 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1
The TSF shall allow **Administrator**-initiated termination of the **Administrator's** own interactive session.

5.2.7.4 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1
Before establishing **an administrative user** session, the TSF shall display **a Security Administrator-specified** advisory **notice and consent** warning message regarding use of the TOE.

5.2.8 Trusted Path/Channels (FTP)

5.2.8.1 FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1
The TSF shall **be capable of using [*SSH, TLS*] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [update server]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **disclosure and detection of modification of the channel data**.

FTP_ITC.1.2
The TSF shall permit **the TSF or the authorized IT entities** to initiate communication via the trusted channel.

FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for *[audit server, update server]*.

5.2.8.2 FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin
The TSF shall **be capable of using [*SSH, TLS, HTTPS*] to** provide a communication path between itself and **authorized** remote **Administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **disclosure and provides detection of modification of the channel data**.

FTP_TRP.1.2/Admin
The TSF shall permit remote **Administrators** to initiate communication via the trusted path.

FTP_TRP.1.3/Admin
The TSF shall require the use of the trusted path for *initial Administrator authentication and all remote administration actions*.

5.2.9 Intrusion Prevention (IPS)

5.2.9.1 IPS_ABD_EXT.1 Anomaly-Based IPS Functionality

IPS_ABD_EXT.1.1 The TSF shall support the definition of [anomaly (‘unexpected’) traffic patterns] including the specification of [

- frequency;]

and the following network protocol fields:

- [
    - *IPv4: IP Flags; Fragment Offset; Protocol; Source Address; Destination Address.*
    - *ICMP: Type; Code.*
    - *TCP: Source port; destination port; TCP flags.*
    - *UDP: Source port; destination port.*
    - *protocol header fields for the following protocols:*
        - *IMAP*
        - *Netbios-ss*]

IPS_ABD_EXT.1.2 The TSF shall support the definition of anomaly activity through [manual configuration by administrators, automated configuration].

IPS_ABD_EXT.1.3 The TSF shall allow the following operations to be associated with anomaly based IPS policies:

- In any mode, for any sensor interface: [
    - allow the traffic flow
    - send a TCP reset to the source address of the offending traffic;
    - send a TCP reset to the destination address of the offending traffic;
    - send an ICMP [host] unreachable message]
- In inline mode:
    - allow the traffic flow
    - block/drop the traffic flow
    - and [no other actions]

5.2.9.2 IPS_IPB_EXT.1 IP Blocking

IPS_IPB_EXT.1.1: The TSF shall support configuration and implementation of known-good and known-bad lists of [source, destination] IP addresses and [*no additional address types*].
IPS_IPB_EXT.1.2: The TSF shall allow [Security Administrators] to configure the following IPS policy elements: [known-good list rules, known-bad list rules, IP addresses, [*ACLs, alert filters*]].

5.2.9.3 IPS_NTA_EXT.1 Network Traffic Analysis

IPS_NTA_EXT.1.1 The TSF shall perform analysis of IP-based network traffic forwarded to the TOE's sensor interfaces and detect violations of administratively-defined IPS policies.
IPS_NTA_EXT.1.2 The TSF shall process (be capable of inspecting) the following network traffic protocols:

- Internet Protocol version 4 (IPv4), RFC 791
- Internet Protocol version 6 (IPv6), RFC 2460
- Internet control message protocol version 4 (ICMPv4), RFC 792
- Internet control message protocol version 6 (ICMPv6), RFC 2463
- Transmission Control Protocol (TCP), RFC 793
- User Data Protocol (UDP), RFC 768

IPS_NTA_EXT.1.3 The TSF shall allow the signatures to be assigned to sensor interfaces configured for promiscuous mode, and to interfaces configured for inline mode, and support designation of one or more interfaces as 'management' for communication between the TOE and external entities without simultaneously being sensor interfaces.

- Promiscuous (listen-only) mode: [*any traffic interface port configured as SPAN*];
- Inline (data pass-through) mode: [*any traffic interface port configured as INLINE*];
- Management mode: [*dedicated management interface*];
- [
    - Session-reset-capable interfaces: [*any traffic interface port configured as INLINE, any traffic interface port configured as RESPONSE*]].

5.2.9.4 IPS_SBD_EXT.1 Signature-Based IPS Functionality

IPS_SBD_EXT.1.1 The TSF shall support inspection of packet header contents and be able to inspect at least the following header 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 [no other field].
- IPv6: Version; payload length; next header; hop limit; source address; destination address; routing header; and [no other field].
- ICMP: Type; Code; Header Checksum; [ID, sequence number, [*other type-specific fields in the ICMP header*]].
- 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.

IPS_SBD_EXT.1.2 The TSF shall support inspection of packet payload data and be able to inspect at least the following data elements to perform string-based pattern-matching:

- ICMPv4 data: characters beyond the first 4 bytes of the ICMP header.
- ICMPv6 data: characters beyond the first 4 bytes of the ICMP header.
- TCP data (characters beyond the 20 byte TCP header), with support for detection of:

    i) FTP (file transfer) commands: 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: commands including GET and POST, and administrator defined strings to match URLs/URIs, and web page content.

    iii) SMTP (email) states: start state, SMTP commands state, mail header state, mail body state, abort state.

    iv) [[*no other types of TCP payload inspection*]];

- UDP data: characters beyond the first 8 bytes of the UDP header;

IPS_SBD_EXT.1.3: The TSF shall be able to detect the following header-based signatures (using fields identified in IPS_SBD_EXT.1.1) at IPS sensor interfaces:

a) IP Attacks

    i) IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack)

    ii) IP source address equal to the IP destination (Land attack)

b) ICMP Attacks

    i) Fragmented ICMP Traffic (e.g. Nuke attack)

    ii) Large ICMP Traffic (Ping of Death attack)

c) TCP Attacks

    i) TCP NULL flags

    ii) TCP SYN+FIN flags

    iii) TCP FIN only flags

    iv) TCP SYN+RST flags

d) UDP Attacks

   i) UDP Bomb Attack
   ii) UDP Chargen DoS Attack

IPS_SBD_EXT.1.4: The TSF shall be able to detect all the following traffic-pattern detection signatures, and to have these signatures applied to IPS sensor interfaces:

a) Flooding a host (DoS attack)
   i) ICMP flooding (Smurf attack, and ping flood)
   ii) TCP flooding (e.g. SYN flood)
b) Flooding a network (DoS attack)
c) Protocol and port scanning
   i) IP protocol scanning
   ii) TCP port scanning
   iii) UDP port scanning
   iv) ICMP scanning

IPS_SBD_EXT.1.5 The TSF shall allow the following operations to be associated with signature-based IPS policies:

- In any mode, for any sensor interface: [
    - allow the traffic flow;
    - send a TCP reset to the source address of the offending traffic;
    - send a TCP reset to the destination address of the offending traffic;
    - send an ICMP [host] unreachable message]
- In inline mode:
    - block/drop the traffic flow;
    - and [
        - allow the traffic flow with following exceptions: [*malicious traffic that matches an attack identified in IPS_SBD_EXT.1.3 or IPS_SBD_EXT.1.4*]]

IPS_SBD_EXT.1.6 The TSF shall support stream reassembly or equivalent to detect malicious payload even if it is split across multiple non-fragmented packets.

5.3 TOE SFR Dependencies Rationale for SFRs

The PP and any relevant EPs/Modules/Packages contain(s) all the requirements claimed in this ST. As such, the dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the PP and any relevant EPs/Modules/Packages, which is/are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in Table 15.

Table 15 – Security Assurance Requirements

Assurance Class	Assurance Components	Component Description
Security Target	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition

Assurance Class	Assurance Components	Component Description
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic functionality specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative Procedures
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_IND.1	Independent testing – conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability survey

5.5 Assurance Measures

The TOE satisfied the identified assurance requirements. This section identifies the Assurance Measures applied by Trellix to satisfy the assurance requirements. The following table lists the details.

Table 16 – TOE Security Assurance Measures

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ALC_CMS.1	
ATE_IND.1	Vendor will provide the TOE for testing.
AVA_VAN.1	Vendor will provide the TOE for testing.

SAR Component	How the SAR will be met
	Vendor will provide a document identifying the list of software and hardware components.

6 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

Table 17 – TOE Summary Specification SFR Description

Requirement	TSS Description
FAU_GEN.1,FAU_GEN_EXT.1	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 13, start-up and shut-down of the audit functions. The type of records generated for each component are determined by Table 12. The audit functions resulting in events being recorded in the auditlog can be enabled/disabled by authorized administrators using the configuration available on Manager. The tracelog auditing is always enabled and cannot be disabled. The following information about an audited event is stored in the audit log whenever that audited event is recorded: • Date and time of the event, • Type (i.e., category and action) of event, • Subject (i.e., user and domain) identity, • Result (success or failure) of the event, and • Description (where applicable access mode, target object, etc.). 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
FAU_GEN.1/IPS	The Sensor(s) generate audit records for the IPS events identified in Table 14 . 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.

Requirement	TSS Description
	For information regarding logging for each field covered by IPS_SBD_EXT.1, refer the corresponding TSS section.
FAU_GEN.2	None
FAU_STG_EXT.1, FAU_STG_EXT.4, FAU_STG_EXT.5	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. These files are transferred securely from the Sensor to the Manager over TLS. 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. If the connection to the syslog server is unavailable, Manager continues to record audit records in the local database; however, any records generated while the syslog server is unavailable will not be transmitted. 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 caches events into 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. No facility is provided on the TOE Sensor to view or modify the audit log file, as the CLI is provided by a zebra shell, which provides no filesystem access and only a limited set of commands and does not support a “root” user. The log file cannot be directly access by any authorized administrator. There is no filesystem access to any administrative users (as the CLI is provided by a zebra shell with a limited set of commands).
FCO_CPC_EXT.1, FPT_ITT.1	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.
FCS_CKM.1	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.

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td></td><td>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.</td></tr>
<tr><td>FCS_CKM.2</td><td>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.

<b>Table 18</b> describes the key establishment schemes and how they are used by the TOE.

<table>
<tr><th>Scheme</th><th>SFR</th><th>Service</th></tr>
<tr><td>ECDHE</td><td>FCS_TLSC_EXT.2
FPT_ITT.1</td><td>ECDHE Intra-TSF Trusted Channel</td></tr>
<tr><td>ECDHE</td><td>FCS_TLSC_EXT.1</td><td>Syslog Server</td></tr>
<tr><td>ECDHE</td><td>FCS_TLSS_EXT.1</td><td>Administration</td></tr>
<tr><td>ECDHE</td><td>FCS_TLSS_EXT.2</td><td>ECDHE Intra-TSF Trusted Channel</td></tr>
<tr><td>ECDH</td><td>FCS_SSHS_EXT.1
FCS_SSHC_EXT.1</td><td>Administration
Importing TOE updates</td></tr>
</table>
<b>Table 18 Key Establishment Schemes</b>

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.</td></tr>
</table>

Requirement	TSS Description
FCS_CKM.4	The TOE meets all requirements specified in FIPS 140-2 for destruction of keys and Critical Security Parameters (CSPs). Refer to Table 20 from section 6.3 for more information on the key zeroization. The TOE does not make use of a value that does not contain any CSP to overwrite keys.
FCS_COP.1/DataEncryption	The TOE performs AES 128- and 256-bit encryption in GCM mode to secure TLS and SSH communication channels.
FCS_COP.1/Hash	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.
FCS_COP.1/KeyedHash	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.
FCS_COP.1/SigGen	The TOE performs RSA 2048-bit SigGen to support TLS functions. The TOE performs RSA 2048 bit SigVer to support TLS, 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 support the SSH public key-based authentication functions (host key).
FCS_HTTPS_EXT.1	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.
FCS_RBG_EXT.1	The TOE implements Counter (CTR) DRBGs, as specified in ISO/IEC 18031:2011 to generate random bits needed for asymmetric key, symmetric key, nonce, and salt generation. **Sensor:** The DRBGs are seeded with 1024 bytes of data from one hardware-based noise source. The 1024 bytes of data contain at least 256-bit of entropy. **Manager:**

Requirement	TSS Description
	IPS Manager DRBG is seeded with 8192 bits or 1024 bytes of data from hardware-based noise source. The 1024 bytes of data contain at least 256-bit of entropy.
FCS_SSHS_EXT.1, FCS_SSHC_EXT.1	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 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 server, 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 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. 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. The SSH Server functionality is used to provide a trusted path for administrative access for Manager as well as Sensors.

Requirement	TSS Description
	If the TOE receives an SSH packet that exceeds 256K, the packet is dropped and logged, and the connection terminated. 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 honors such an attempt.
FCS_TLSC_EXT.1	The TOE (Manager) operates as TLS Client without mutual authentication 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 The TOE presents the Supported Elliptic Curves Extension indicating support for P-256 and P-384 in the Client Hello. 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.
FCS_TLSC_EXT.2, FCS_TLSS_EXT.2, FPT_ITT.1	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.
FCS_TLSS_EXT.1	The TOE acts as a TLS/HTTPS server without mutual authentication 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

Requirement	TSS Description
	Key Establishment is performed using Elliptic Curve Diffie-Hellman P-256 keys. The TOE supports session resumption using session IDs, in accordance with RFC 5246. 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.
FIA_AFL.1, FIA_UIA_EXT.1 , FIA_UAU_EXT.2, FIA_UAU.7 , FIA_PMG_EXT.1, FTA_TAB.1	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). 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. 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. 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 ("!", "@", "#", "$", "%", "^", "&", "*", "(", ")"). The enforced Manager CLI password requirements are as follows: • Minimum length of 15 characters. • Contains at least 1 lowercase, 1 upper case, 1 numeric and 1 special character ("!", "@", "#", "$", "%", "^", "&", "*", "(", ")").

Requirement	TSS Description
	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 “/”). 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 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. Sensors also support a CLI interface using SSH with password and public key authentication. This interface is used for initial setup and registration of the appliances. Once registration with the Manager is complete all management of the Sensors can be performed via the Manager.
FIA_X509_EXT.1/Rev , FIA_X509_EXT.1/ITT , FIA_X509_EXT.2, FIA_X509_EXT.3	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

Requirement	TSS Description
	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. Only the Manager component can generate CSRs for the TOE 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.
FMT_MOF.1/ManualUpdate , FMT_MTD.1/CoreData, FMT_SMR.2	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, which is the only functionality available prior to the administrator login. This TOE banner is administrator configurable. Having acknowledged the access banner, the user is then prompted to enter their username and password. The TOE CLI and web GUI provide a security management interface used to configure and manage the TOE. All management of the TOE is either performed through the Manager, which then pushes sensor-related configuration updates to the Sensor(s), or directly on the Sensor, which thereafter generates appropriate logs on the Manager. 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. Local administrator access is provided by the keyboard/monitor interface on the Manager, and console port on the Sensors, while remote administrator access is provided via SSH or TLS/HTTPS.
FMT_SMF.1, FMT_SMF.1/IPS	The Security Administrator is authorized to: • administer the TOE locally and remotely; • configure the access banner; • configure the session inactivity timeout; • update the TOE and verify the updates using digital signature; • configure the authentication failure parameters; • configure audit behaviour; • modify the behaviour of the transmission of audit data to an external IT entity; • configure the cryptographic functionality;

Requirement	TSS Description
	• configure thresholds for SSH rekeying; • configure the interaction between TOE components; • re-enable an Administrator account; • set the time which is used for time-stamps; • manage the TOE's trust store and designate X509.v3 certificates as trust anchors; • import X.509v3 certificates to the TOE's trust store; • manage the trusted public keys database. The Security Administrator is also authorized to perform the following IPS management functions: • enable, disable signatures applied to sensor interfaces, and determine the behavior of IPS functionality; • modify the following 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. Local administrator access is provided by the keyboard/monitor interface on the Manager, and console port on the Sensors, while remote administrator access is provided via SSH or TLS/HTTPS. 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)

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

All IPS-related management functions are accessible only via the remote GUI interface.
IPS data analysis and reactions can be configured as detailed in the IPS_ABD_EXT.1, IPS_IPB_EXT.1 & IPS_SBD_EXT.1 TSS sections.
Interaction between the Trellix IPS components i.e. Manager and sensors can be configured and managed as explained in the FCO_CPC_EXT.1 TSS section.
</td></tr>
</table>

Requirement	TSS Description
FPT_APW_EXT.1, FPT_SKP_EXT.1	There is no filesystem access or administrative interface that allows any administrative users to read plaintext pre- shared keys, symmetric keys, and private keys. Table 20 and Table 21 elaborate on how these keys are stored. 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.
FPT_STM_EXT.1	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.
FPT_TST_EXT.1	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

Requirement	TSS Description
	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 performs 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. These tests, which are performed separately on both the sensor as well as Manager, are sufficient to demonstrate the TSF is operating correctly, as they confirm the integrity of all modules prior to their installation thereby confirming the modules have not been modified or replaced in any unauthorized manner. They also ensure the DRNG & NDRNG continue to operate successfully providing sufficient entropy in response to any requests. Lastly the cryptographic self-test ensure accurate operation of all supported algorithms.
FPT_TUD_EXT.1	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.

Requirement	TSS Description
FTA_SSL.3, FTA_SSL.4, FTA_SSL_EXT.1	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. 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.
FTP_ITC.1, FTP_TRP.1/Admin	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 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. The TOE provides the following version and algorithm support: • TLS Version: v1.2 • Proposed Cipheruites: ○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 The TOE (Manager) also acts as a TLS/HTTPS server to provide management via web GUI to administrators. This server supports TLSv1.2 with the following ciphersuites: • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Requirement	TSS Description
	• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 Key Establishment is performed using Elliptic Curve Diffie-Hellman P-256 keys.
IPS_ABD_EXT.1	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.
IPS_IPB_EXT.1	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.
IPS_NTA_EXT.1	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. 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

Requirement	TSS Description
	IPS_IPB_EXT.1 can be manually re-adjusted by the administrator even after definition. 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.
IPS_SBD_EXT.1	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. 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. 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. For all types of signature-based attacks with a blocking or quarantine action, the TOE logs and also as maintains a counter of the packets dropped by it.

6.1 Distributed TOE SFR Allocation

For a distributed TOE, the SFR in the PP as well as any relevant EPs/Modules/Packages must be met by the TOE as a whole. However, each TOE component will not necessarily meet each SFR. Table specifies when each SFR must be implemented by a component. Since SFRs drawn from [MOD_IPS] are not addressed by the distribution requirements in NDcPP, their distribution requirements in the following table have been specified to be consistent with the requirements for similar SFRs.

The following categories are used to define those SFR allocations:

- All Components (All): All components that comprise of the distributed TOE must independently satisfy the requirement.
- At least one Component (One): This requirement must be fulfilled by at least one component within the distributed TOE.
- Feature Dependent (Feature Dependent): These requirements will only be fulfilled where the feature is implemented by the distributed TOE component.

Table 19– Distributed TOE SFR Allocation

Requirement	SFR Allocation	Manager	Sensor	Rationale
FAU_GEN.1	All	Y	Y	Satisfied
FAU_GEN.1/IPS	Feature Dependent	N	Y	Satisfied since only the Sensors generate the audits required by this SFR.
FAU_GEN.2	All	Y	Y	Satisfied
FAU_GEN_EXT.1	All	Y	Y	Satisfied
FAU_STG_EXT.1	Feature Dependent	Y	N	Satisfied since only the Manager stores audits.
FAU_STG_EXT.4	Feature Dependent	Y	N	Satisfied since only the Manager stores audits locally.
FAU_STG_EXT.5	Feature Dependent	N	Y	Satisfied since only the Sensor stores audits remotely.
FCO_CPC_EXT.1	All	Y	Y	Satisfied.
FCS_CKM.1	One	Y	Y	Satisfied
FCS_CKM.2	All	Y	Y	Satisfied
FCS_CKM.4	All	Y	Y	Satisfied
FCS_COP.1/ DataEncryption	All	Y	Y	Satisfied
FCS_COP.1/SigGen	All	Y	Y	Satisfied
FCS_COP.1/Hash	All	Y	Y	Satisfied
FCS_COP.1/KeyedHash	All	Y	Y	Satisfied
FCS_HTTPS_EXT.1	Feature Dependent	Y	N	Satisfied since only the Manager provides HTTPS functionality.
FCS_RBG_EXT.1	All	Y	Y	Satisfied

Requirement	SFR Allocation	Manager	Sensor	Rationale
FCS_SSHS_EXT.1	Feature Dependent	Y	Y	Satisfied.
FCS_SSHC_EXT.1	Feature Dependent	N	Y	Satisfied
FCS_TLSC_EXT.1	Feature Dependent	Y	N	Satisfied since only the Manager provides TLS client functionality without mutual authentication.
FCS_TLSC_EXT.2	Feature Dependent	N	Y	Satisfied since only the Sensors provide TLS client functionality with mutual authentication.
FCS_TLSS_EXT.1	Feature Dependent	Y	N	Satisfied since only the Manager provides TLS functionality without mutual authentication.
FCS_TLSS_EXT.2	Feature Dependent	Y	N	Satisfied since only the Manager provides TLS server functionality with mutual authentication.
FIA_AFL.1	One	Y	Y	Satisfied
FIA_PMG_EXT.1	One	Y	Y	Satisfied
FIA_UIA_EXT.1	One	Y	Y	Satisfied
FIA_UAU_EXT.2	One	Y	Y	Satisfied since only the Manager provides I&A functionality for local admins.
FIA_UAU.7	Feature Dependent	Y	Y	Satisfied
FIA_X509_EXT.1/ITT	Feature Dependent	Y	Y	Satisfied
FIA_X509_EXT.1/Rev	Feature Dependent	Y	N	Satisfied since only the Manager provides revocation checking for X509 certificates functionality.
FIA_X509_EXT.2	Feature Dependent	Y	N	Satisfied
FIA_X509_EXT.3	Feature Dependent	Y	N	Satisfied since only the Manager provides CSR generation functionality.
FMT_MOF.1/ ManualUpdate	All	Y	Y	Satisfied
FMT_MTD.1/CoreData	All	Y	Y	Satisfied
FMT_SMF.1	Feature Dependent	Y	Y	Satisfied
FMT_SMF.1/IPS	Feature Dependent	Y	Y	Satisfied
FMT_SMR.2	One	Y	Y	Satisfied
FPT_APW_EXT.1	Feature Dependent	Y	Y	Satisfied

Requirement	SFR Allocation	Manager	Sensor	Rationale
FPT_ITT.1	Feature Dependent	Y	Y	Satisfied
FPT_SKP_EXT.1	All	Y	Y	Satisfied
FPT_STM_EXT.1	All	Y	Y	Satisfied
FPT_TST_EXT.1	All	Y	Y	Satisfied
FPT_TUD_EXT.1	All	Y	Y	Satisfied
FTA_SSL_EXT.1	Feature Dependent	Y	Y	Satisfied
FTA_SSL.3	Feature Dependent	Y	Y	Satisfied
FTA_SSL.4	Feature Dependent	Y	Y	Satisfied
FTA_TAB.1	All	Y	Y	Satisfied
FTP_ITC.1	All	Y	Y	Satisfied
FTP_TRP.1/Admin	One	Y	Y	Satisfied
IPS_ABD_EXT.1	Feature Dependent	N	Y	Satisfied since only the Sensors provide IPS functionality.
IPS_IPB_EXT.1	Feature Dependent	N	Y	Satisfied since only the Sensors provide IPS functionality.
IPS_NTA_EXT.1	Feature Dependent	N	Y	Satisfied since only the Sensors provide IPS functionality.
IPS_SBD_EXT.1	Feature Dependent	N	Y	Satisfied since only the Sensors provide IPS functionality.

6.2 Cryptographic Key Destruction

The table below describes the key zeroization provided by the TOE and as referenced in FCS_CKM.4 for Manager.

Table 20 - Manager Key zeroisation

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
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	Plaintext, Loaded to the disk and into RAM	N/A – Public Key

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
	Manager firmware images		

The following table specifies the zeroization method of cryptographic keys generated and managed on the Sensor.

Table 21 - Sensor Key zeroisation

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) and are enabled by default.	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.
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	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.	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
		"admin" access and are configurable via CLI (privatemode enable|disable) and are enabled by default.		The "support" and "private" passwords are stored via HMAC-SHA-512 hash in shell.conf file.		
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. Output: Only displayed in the private mode command.	Zeroized from RAM on each reboot.
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.

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
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
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-	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

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
		256/512) keys created for each SCP session.				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 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	Zeroized as part of openSSL scrubbing and in RAM on reboot

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
					Output: Never output	
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
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
21	SSH Session Public Key	EC Diffie-Hellman P-256-bit session	Internally using the Block Cipher	Temporarily in RAM	Entry: Never entered	Zeroized in RAM on reboot and

Sr No.	Key	Description/ Usage	Generation	Storage	Entry/Output	Destruction
	(Sensor as Client)	key created for each SCP session	(CTR) DRBG provided by OpenSSL	stored as Plaintext	Output: Never output	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.

7 Acronym Table

Acronyms should be included as an Appendix in each document.

Table 22 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CC	Common Criteria
CRL	Certificate Revocation List
DTLS	Datagram Transport Layer Security
EP	Extended Package
GUI	Graphical User Interface
IP	Internet Protocol
NDcPP	Network Device Collaborative Protection Profile
NIAP	Nation Information Assurance Partnership
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
PP	Protection Profile
RSA	Rivest, Shamir & Adleman
SFR	Security Functional Requirement
SigGen	Signature Generation
SigVer	Signature Verification
SSH	Secure Shell
SSH KDF	SSH Key Derivation Function
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSS	TOE Summary Specification
TSF	TOE Security Functionality