Trend Micro TippingPoint Threat Protection System (TPS) v5.5.0 Security Target

Version 1.0
May 22, 2023

Prepared for:

TREND MICRO

11305 Alterra Parkway
Austin, TX 78758

Prepared by:

leidos

Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive, Columbia, Maryland 21046

LIST OF TABLES

1. Security Target Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE references, TOE overview, and TOE description. It also contains the ST and TOE conformance claims, ST conventions, glossary, and list of abbreviations. The TOE is the Trend Micro TippingPoint Threat Protection System TPS) v5.5.0 provided by Trend Micro. The Trend Micro product is a network security solution for advanced threat detection.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Trend Micro TPS offers a series of documents that describe the installation process for the TOE, as well as guidance for subsequent use and administration of the system security features. The following documents are available for download from the Trend Micro Online Help Center: https://docs.trendmicro.com/en-us/tippingpoint/threat-protection-system.aspx.
- Trend Micro TippingPoint Threat Protection System Hardware Specification and Installation Guide, September 2020
- Trend Micro TippingPoint Threat Protection System Command Line Interface Reference, July 2021
- Trend Micro TippingPoint Virtual Threat Protection System (vTPS) User Guide, October 2021
- Trend Micro TippingPoint Threat Protection System Common Criteria Evaluated Configuration Guide (CCECG) for TPS v5.5.0, Document Version 1.0, 22 May 2023
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- Development (ADV)

ADV_ARC.1 – Security architecture description

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

ADV_ARC.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2 – Security-enforcing functional specification

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.2D The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.2.1C The functional specification shall completely represent the TSF.

ADV_FSP.2.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4C For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.2.5C For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

ADV_TDS.1 – Basic design

ADV_TDS.1.1D The developer shall provide the design of the TOE.

ADV_TDS.1.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.1.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.1.2C The design shall identify all subsystems of the TSF.

ADV_TDS.1.3C The design shall describe the behaviour of each SFR-supporting or SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.

ADV_TDS.1.4C The design shall summarise the SFR-enforcing behaviour of the SFR-enforcing subsystems.

ADV_TDS.1.5C The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.

ADV_TDS.1.6C The mapping shall demonstrate that all TSFIs trace to the behaviour described in the TOE design that they invoke.

ADV_TDS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.1.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

1.1.1 Guidance Documents (AGD)

AGD_OPE.1 – Operational user guidance

AGD_OPE.1.1D The developer shall provide operational user guidance.

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1 – Preparative procedures

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

1.1.2 Life-cycle (ALC)

ALC_CMC.2 – Use of a CM system

ALC_CMC.2.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.2.2D The developer shall provide the CM documentation.

ALC_CMC.2.3D The developer shall use a CM system.

ALC_CMC.2.1C The TOE shall be labelled with its unique reference.

ALC_CMC.2.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.2.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_CMS.2 – Parts of the TOE CM coverage

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

ALC_CMS.2.1C The configuration list shall include the following: The TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.

ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.2.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

ALC_CMS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DEL.1 – Delivery procedures

ALC_DEL.1.1D The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

ALC_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

1.1.3 Security Target Evaluation (ASE)

ASE_CCL.1 – Conformance claims

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

ASE_CCL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1 – Extended components definition

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

ASE_ECD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

ASE_INT.1 – ST introduction

ASE_INT.1.1D The developer shall provide an ST introduction.

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

ASE_INT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

ASE_OBJ.2 – Security objectives

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

ASE_OBJ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_REQ.2 – Derived security requirements

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

ASE_REQ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_SPD.1 – Security problem definition

ASE_SPD.1.1D The developer shall provide a security problem definition.

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

ASE_SPD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1 – TOE summary specification

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

ASE_TSS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

1.1.4 Tests (ATE)

ATE_COV.1 – Evidence of coverage

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 – Functional testing

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 – Independent testing - sample

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

1.1.5 Vulnerability Assessment (AVA)

AVA_VAN.2 – Vulnerability analysis

AVA_VAN.2.1D The developer shall provide the TOE for testing.

AVA_VAN.2.1C The TOE shall be suitable for testing.

AVA_VAN.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

- TOE Summary Specification (Section 5.3.1)
-

- (Section 1)
- Rationale (Section 7).

1.2 Security Target, TOE and CC Identification

ST Title – Trend Micro TippingPoint Threat Protection System (TPS) v5.5.0 Security Target

ST Version – 1.0

ST Date – May 22, 2023

TOE Identification – Trend Micro TippingPoint Threat Protection System (TPS) v5.5.0

The TOE consists of the following appliances running TPS software v5.5.0:

- Trend Micro TippingPoint 1100TX (TPNN0321)
- Trend Micro TippingPoint 5500TX (TPNN0322)
- Trend Micro TippingPoint 8200TX (TPNN0090)
- Trend Micro TippingPoint 8400TX (TPNN0091)
- Trend Micro TippingPoint vTPS (VMware) (vTPS_vmw_5.5.0.2130.zip)
- Trend Micro TippingPoint vTPS (KVM) (vTPS_ kvm_5.5.0.2130.tar.gz)

Hardware Appliance Device Model	Trend Micro part number
TippingPoint 1100TX	TPNN0321
TippingPoint 5500TX	TPNN0322
TippingPoint 8200TX	TPNN0090
TippingPoint 8400TX	TPNN0091

The 1100TX includes one I/O module slot, the 5500TX includes two I/O module slots, and the 8200TX and the 8400TX include four I/O module slots. The following standard I/O modules are supported for the 1100TX, 5500TX, 8200TX, and 8400TX security devices.

Standard I/O module	Trend Micro part number
TippingPoint 6-Segment Gig-T	TPNN0196/TPNN0059
TippingPoint 6-Segment GbE SFP	TPNN0068
TippingPoint 4-Segment 10 GbE SFP+	TPNN0060
TippingPoint 1-Segment 40 GbE QSFP+	TPNN0069

The vTPS virtual appliances consist of TPS v5.5.0, running on hosts with Intel Haswell-based or Ivy Bridge-based microprocessors and either

- an ESXi Hypervisor: Version 6.7 or 7.0.2 (only paid versions supported), or
- a RHEL version 7.1 KVM.

The vTPS virtual appliances use virtual data ports and do not require I/O modules.

The vTPS appliances are provided as image files:

- vTPS_vmw_5.5.0.2130.zip
- vTPS_kvm_5.5.0.2130.tar.gz

1.3 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 5, April 2017.
  - Part 3 Conformant

This ST and the TOE it describes are conformant to the following package:

- EAL2

1.4 Conventions

The following conventions are used in this document:

Extended requirements – Security Functional Requirements not defined in Part 2 of the CC are annotated with a suffix of _EXT.

Security Functional Requirements – Part 1 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.

- ○ Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").
- ○ Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
- ○ Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
- ○ Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note that 'cases' that are not applicable in a given SFR have simply been removed without any explicit identification.

Other sections of the ST – Other sections of the ST use bolding and/or different fonts (such as `Courier`) to highlight text of special interest, such as captions.

1.4.1 Terminology

This section identifies TOE-specific terminology.

DV	Digital Vaccine
HA	High Availability
IPM	Trend Micro's proprietary IP Protection Module
ISO	An ISO image (or .ISO file) is a computer file that is an exact copy of an existing file system
LSM	Local Security Manager
SMS	Security Management System
TPS	TippingPoint Threat Protection System (the TOE)

1.4.2 Abbreviations

This section identifies abbreviations and acronyms used in this ST.

AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher-Block Chaining
CA	Certificate Authority
CLI	Command Line Interface
CM	Configuration Management
DH	Diffie-Hellman
FIPS	Federal Information Processing Standard
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
ICAP	Internet Content Adaptation Protocol
NDPP	Protection Profile for Network Devices
NIST	National Institute of Standards and Technology
OS	Operating System
RSA	Rivest, Shamir and Adleman (algorithm for public-key cryptography)
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SMB	Server Message Block
SNMP	Simple Network Management Protocol
SSD	Solid State Drive
SSH	Secure Shell
SSL	Secure Socket Layer
ST	Security Target
TCP	Transmission Control Protocol
TOE	Target of Evaluation
TSF	TOE Security Functions
UAU	User Authentication
UDP	User Datagram Protocol
VM	Virtual Machine

2. TOE Description

The Target of Evaluation (TOE) is the Trend Micro TippingPoint Threat Protection System (TPS) v5.5.0. It may also be referred to as TippingPoint Threat Protection System or simply TPS. TPS is a network security platform that offers threat protection, shielding network vulnerabilities, blocking exploits, and defending against known and zero-day attacks (Intrusion Prevention System (IPS) capabilities). TPS provides coverage across various threat vectors, including advanced threats, malware, and phishing attempts. It employs a combination of technologies, such as deep packet inspection, threat reputation, and malware analysis, on a flow-by-flow basis, in order to detect and prevent attacks on the network. The product consists of the Threat Suppression Engine (TSE), Traffic Management filters, and Digital Vaccine (DV) filters that provide threat protection, shielding network vulnerabilities, blocking exploits, and defending against known and zero-day attacks. The TOE's specialized hardware-based traffic classification engines enable the IPS to filter accurately at gigabit speeds and microsecond latencies. Unlike software-based systems whose performance may be affected by the number of filters installed, the scalable capacity of the TOE's hardware engine allows thousands of filters to run simultaneously with no impact on performance or accuracy. The TOE was evaluated as an IPS network device and does not include evaluation of the aforementioned speed or latency claims.

The TPS version 5.5.0 appliances included in the evaluation are TPS 1100TX, TPS 5500TX, TPS 8200TX, TPS 8400TX, and vTPS. Each physical appliance includes an RJ-45 console port and a 1 GbE copper management port. The 8200TX and 8400TX devices are high-end systems that are designed for network environments requiring up to 40 Gbps of inspection throughput. The 1100TX and 5500TX devices support the same I/O modules as the 8200TX and 8400TX so these models can support the same capacity on a per-module basis, but they have fewer module slots for a reduced overall performance capacity. The concept of IO modules is not applicable to the vTPS model which has two virtual data ports.

The vTPS model is a virtual appliance supported on VMware and KVM. Each virtual platform supports a virtual serial console and virtual Ethernet management port. Each virtual appliance deployed in normal mode provides 500 Mbps IPS inspection throughput with two vCPUs or 1 Gbps IPS inspection throughput with three vCPUs. When deployed in Performance mode, six vCPUs provide 2 Gbps IPS inspection throughput. Each vTPS supports one vNIC (VMware) or one bridge interface (KVM) for management.

All models (hardware and virtual) provide the same security protections and support all of the functionality specified in this ST.

The TOE uses NIST validated cryptographic algorithms and must be configured to operate in FIPS mode in order to use them.

2.1 TOE Overview

The TippingPoint Threat Protection System v5.5.0 is a network device with threat protection services provided as a standalone hardware or virtual appliance. The appliances include the TPS 5.5.0 software.

Each appliance also includes the hardened Linux-4.14.76-yocto-standard operating system. All hardware models include external user disk memory (CFast or SSD) that is used to store all traffic logs, snapshots, ThreatDV URL Reputation Feed, User-defined URL Entries database, and packet capture data. The external memory can also be used for troubleshooting purposes. vTPS appliances do not have a separate user disk. The vTPS virtual appliances have a single-disk architecture with either an 8-GB user disk partition (for standard) or 16-GB user disk partition (for Performance). The TX hardware models include standard I/O modules used to receive and transmit packets for the threat detection functions. The 1100TX includes one I/O module slot, the 5500TX includes two I/O module slots, and the 8200TX and the 8400TX include four I/O module slots. The supported standard I/O modules are identified in Section 1.1. The concept of IO modules is not applicable to vTPS which has two virtual data ports.

The TOE provides intrusion prevention services including monitoring, collection, inspection, analyzation, and reaction capabilities applied to network traffic in real-time. The TOE reconstructs and inspects flow payloads by parsing the traffic at the application layer. As each new packet of the traffic flow arrives, the engine re-evaluates the traffic for malicious content. The instant the engine detects malicious traffic, it blocks all current and all subsequent packets pertaining to the traffic flow. The blocking of the traffic and packets ensures that the attack never reaches its destination. The TOE provides authorized administrators with a CLI accessible via SSH to manage the TOE and its

IPS functions and to monitor, collect, log, and react in real-time to potentially malicious network traffic. Evaluation of the IPS services focuses on inspecting the IPv4 and IPv6 traffic (TCP, UDP, ICMP, etc.).

The TOE requires users to be identified and authenticated before they can access any of the TOE functions. For each session, the user is required to log in prior to successfully establishing a session through which TOE functions can be exercised. The only capabilities allowed prior to users authenticating are the display of the warning banner before authentication, and the TOE may send Echo Reply in response to Echo Request ICMP messages received at the Management interface. The banner is displayed on every login attempt.

The authorized administrators interact locally with the TOE via console or remotely using SSH where OpenSSL is used to implement SSH and its underlying core cryptographic algorithms to secure the underlying communications. The TOE also uses SSH for communications with trusted external syslog servers. The TOE is operated in FIPS mode and includes NIST validated cryptographic algorithms.

The TOE local and remote administration is provided through the Command Line Interface (CLI). The TOE supports Super User, Admin, and Operator roles that collectively represent the Security Administrator role described in the Security Problem Definition and the Security Functional Requirements. Each user must be assigned a role in order to perform any management action.

The TOE can communicate with the Trend Micro website to download TOE updates. The management CLI provided by the TOE can be used by Super User or Admin administrators to update the TOE, and to query the currently executing software version of the TOE. Software updates are available as package files. The update package is published on Trend Micro support website and protected with a SHA-256 hash, and signed using 2048-bit RSA public/private key pair.

The TOE audit log provides an internal log implementation that can be used to store and review audit records locally. Access is available to the Super User. The TOE can also be configured to send generated audit records to an external Syslog server using SSH. When configured to send audit records to a syslog server, audit records are written to the external syslog as they are written locally to the TOE Audit log.

A sample deployment scenario is as follows.

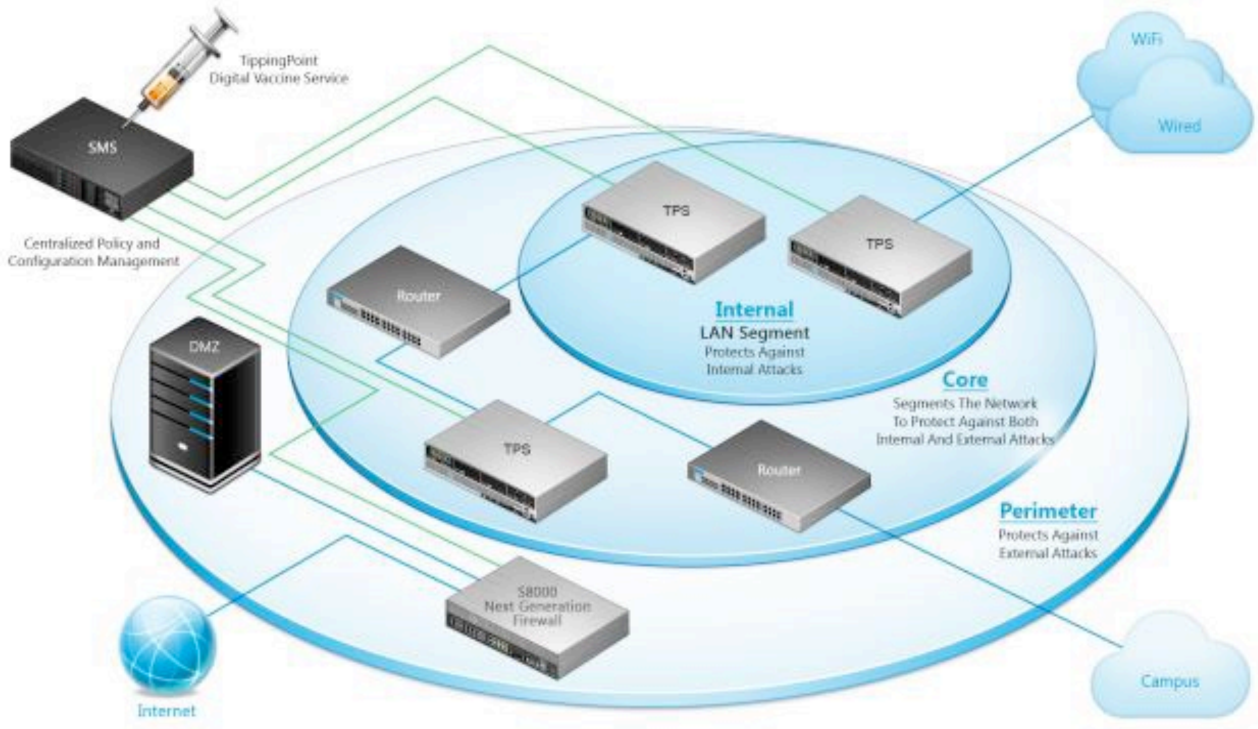

TippingPoint Digital Vaccine Service
SMS
Centralized Policy and Configuration Management
DMZ
Router
TPS
TPS
Internal
LAN Segment
Protects Against Internal Attacks
Core
Segments The Network To Protect Against Both Internal And External Attacks
TPS
Router
Perimeter
Protects Against External Attacks
S8000 Next Generation Firewall
Internet
WiFi
Wired
Campus

Figure 1 – Sample TPS Network Deployment Scenario

Figure 1 Sample TPS Network Deployment Scenario depicts an example of a corporate network with the TPS deployed to a variety of locations. A single TPS can be installed at the perimeter of the network, at the network core, on your intranet, or in all three locations. Though not depicted in the figure, the evaluated deployment includes a syslog server. The SMS appliance may be used in the evaluated configuration however it is not covered by the evaluation and no claims about its behavior are made. Note that the TOE is evaluated as a single appliance, not a distributed solution. A single appliance is sufficient for the TOE to address the claimed security functionality. Deployment of multiple separate instances of the TOE is necessary only to ensure full coverage of network segments that require monitoring.

vTPS appliances are deployed to appropriate hardware that are located between L2 broadcast domains (VLANs or switches).

2.2 TOE Architecture

This section describes the TOE physical and logical boundaries.

2.2.1 Physical Boundaries

The TOE is a self-contained hardware appliance or VM with TPS 5.5.0 software.

The following table identifies the hardware appliance models included in the TOE.

Device	Main Processor	Storage	Network Ports	Operating System / Software
TPS1100TX	Intel Pentium D-1517 (Broadwell with AES-NI) CPU / 4 Cores, 8 Threads, 1.6GHz, 25W TDP	Storage = 8GB CFAST (Internal) / 8GB (External)	One IOM Slot Hot-Swappable Up to 6 1GE Segments, Up to 4 10GE Segments, 1 40GE Segment	Linux-4.14.76-yocto-standard OpenSSL 1.0.2l-fips
TPS5500TX	Intel Xeon D-1559 (Broadwell with AES-NI) CPU / 12 Cores, 24 Threads, 1.5GHz, 45W TDP	Storage = 32GB CFAST (Internal) / 32GB (External)	Two IOM Slots, Hot-Swappable Up to 12 1GE Segments, Up to 8 10GE Segments, Up to 2 40GE Segments	Linux-4.14.76-yocto-standard OpenSSL 1.0.2l-fips
TPS8200TX	2x Intel Xeon E5-2648Lv3 (with AES-NI) CPUs / 12 Cores, 24 Threads, 1.8GHz, 75W TDP	Storage = 32GB CFAST (Internal) / 32GB (External)	Four IOM Slots, Two Hot-Swappable Up to 12 1GE Segments, Up to 8 10GE Segments, Up to 2 40GE Segments	Linux-4.14.76-yocto-standard OpenSSL 1.0.2l-fips
TPS 8400TX	2x Intel Xeon E5-2648Lv3 (with AES-NI) CPUs / 12 Cores, 24 Threads, 1.8GHz, 75W TDP	Storage = 32GB CFAST (Internal) / 32 GB (External)	Four IOM Slots, Hot-Swappable Up to 24 1GE Segments, Up to 16 10GE Segments, Up to 4 40GE Segments	Linux-4.14.76-yocto-standard OpenSSL 1.0.2l-fips

Table 1 TOE Hardware Appliances

The TippingPoint vTPS is deployed between layer 2 (L2) broadcast domains (virtual switches) using an image with either "Normal" or "Performance" options. Performance option offers an increased capacity for vCPUs and threading.

Virtual Machine appliance TOEs consist of TPS v5.5.0, including Linux-4.14.76-yocto-standard and OpenSSL 1.0.2l-fips and requires the following:

Device	Image	Number of vCPUs	Memory	Disk	Operating System / Software
vTPS	Normal Option: • VMware: vTPS_vmw_5.5.0_xxxx.zip Or • KVM: vTPS_kvm_5.5.0_xxxx.tar.gz	2 – 3	8GB	16.2GB	ESXi Hypervisor version: Version 6.7 or 7.0.2 (only paid versions supported) or RHEL version 7.1 KVM
	Performance Option: • VMware: vTPS_vmw_5.5.0_xxxx.zip Or • KVM: vTPS_kvm_5.5.0_xxxx.tar.gz	6	16GB	16.2GB	ESXi Hypervisor version: Version 6.7 or 7.0.2 (only paid versions supported) or RHEL version 7.1 KVM

Table 2 TOE Virtual Machine Appliances

vTPS virtual appliances are supported on hosts with Intel Haswell-based or Ivy Bridge-based microprocessors.

2.2.1.1 Software Requirements

The TOE virtual (VM) appliances are delivered as an installation disk (or ISO image). They require that the following are installed on the host hardware system:

- VMware ESXi 6.7 or 7.0.2 (only paid versions supported)
- RHEL version 7.1 KVM

2.2.1.2 Additional Hardware Requirements

- External audit storage requires the use of syslog servers.
- An administrative workstation or terminal emulator equipped with SSH client software.

2.2.1.3 Exclusions

The TippingPoint Threat Protection System solution includes Local Security Management (LSM) and Security Management System (SMS) components that provides remote administrative management. In the evaluated configuration, all management must be performed using the CLI.

The Digital Vaccine service is provided by the TOE developer and assumed to be a trusted service. It may be used in the evaluated configuration; however it is not included in the TOE itself and therefore no claims are made about its ability to provide adequate or timely filter updates.

The TPS devices can be configured to use sFlow record emission to sample a random flow of traffic and send the data to a collector server for analysis. SFlow and collector services are excluded from the evaluated configuration and must not be configured or used.

Two TippingPoint Threat Protection appliances can be installed in a redundant network configuration. This system configuration provides High Availability (HA), ensuring that the network traffic always flows at wire speeds in the event of any internal hardware or software failure on the device. HA configurations are not covered in the scope of the evaluation.

TippingPoint Threat Protection appliances can be installed in a stacking configuration. Stacking enables an organization to increase the overall inspection capacity of the TPS by grouping multiple TX Series devices and pooling their resources. Stacking configurations are not included in the evaluated configuration. The devices are being evaluated in a standalone configuration.

Optional bypass I/O modules are available for the 1100TX, 5500TX, 8200TX, and 8400TX security devices that provide high availability for copper and fiber segments. These modules are not included in the TOE and must not be used in the evaluated configuration.

2.2.2 Logical Boundaries

This section summarizes the security functions provided by the TOE:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels
- Intrusion Prevention System

2.2.2.1 Security audit

The TOE is able to generate audit records for security relevant events including IPS-related events. The TOE can be configured to store the audit records locally on the TOE and can also be configured to send the logs to a designated external log server. The audit records in local audit storage cannot be modified or deleted. In the event the space available for storing audit records locally is exhausted, the TOE deletes the oldest historical log file, renames the current log file to be a historical file, and creates a new current log file. The TOE will write a warning to the audit trail when the space available for storage of audit records exceeds 75% space remaining threshold.

2.2.2.2 Cryptographic support

The TOE is operated in FIPS mode and includes FIPS-approved and NIST-recommended cryptographic algorithms. The TOE provides cryptographic mechanisms for symmetric encryption and decryption, cryptographic signature services, cryptographic hashing services, keyed-hash message authentication services, deterministic random bit generation seeded from a suitable entropy source, and key zeroization/destruction. The cryptographic mechanisms support SSH used for secure communication, both as client and server.

2.2.2.3 Identification and authentication

The TOE requires users (i.e., administrators) to be successfully identified and authenticated before they can access any security management functions available in the TOE. The TOE offers both a locally connected console and a network accessible interface over SSH to support administration of the TOE.

The TOE supports the local (i.e., on device) definition of administrators with usernames and passwords. When a user is authenticated at the local console, no information about the authentication data (i.e., password) is echoed to the user. Passwords can be composed of any combination of upper and lower case letters, numbers, and the following special characters: !; @; #; $; %; ^; &; *; (;); ,; .; ?; <; >; and /.

The TOE provides authentication failure handling for remote administrator access. When the defined number of unsuccessful authentication attempts has been reached, the remote administrator accessing the TOE via SSH is locked out for an administrator configurable period of time. Authentication failures by remote administrators cannot lead to a situation where no Administrator access is available to the TOE since administrator access is still available via local console.

2.2.2.4 Security management

The TOE provides administrator roles and supports local and remote administration. The TOE supports Super User, Admin, and Operator roles that together comprise the Security Administrator role. Each user must be assigned a role in order to perform any management action. The TOE provides authorized administrators with a CLI accessible via SSH or locally through the console interface for TOE configuration and to monitor, collect, log, and react in real-time to potentially malicious network traffic.

2.2.2.5 Protection of the TSF

The TOE protects sensitive data such as stored passwords and cryptographic keys so that they are not accessible even by an administrator. It also provides its own timing mechanism that ensures reliable time information is available.

The TOE provides mechanisms to view the current version of the TOE and to install updates of the TOE software. TOE updates are initiated manually by the Super User or Admin, who can verify the integrity of the update prior to installation using a digital signature.

The TOE performs tests for software module integrity and cryptographic known-answer tests.

2.2.2.6 TOE access

The TOE implements administrator-configurable session inactivity limits for local interactive sessions at the console and for SSH sessions. The TOE will terminate such sessions when the inactivity period expires. In addition, administrators can terminate their own interactive sessions by logging out at the console and SSH.

The TOE supports an administrator-configurable TOE access banner that is displayed prior to a user completing the login process at the CLI. This is implemented for both local and remote management connections (console, SSH).

2.2.2.7 Trusted path/channels

The TOE protects interactive communication with remote administrators using SSH. SSH ensures confidentiality of transmitted information and detects any loss of integrity.

The TOE also uses SSH to protect the transmission of audit records to an external audit server

2.2.2.8 Intrusion Prevention System

The TOE provides intrusion prevention services including collection, inspection, analyzation, and reaction capabilities applied to network traffic in real-time.

2.3 TOE Documentation

Trend Micro TPS offers a series of documents that describe the installation process for the TOE, as well as guidance for subsequent use and administration of the system security features. The following documents are available for download from the Trend Micro Online Help Center: https://docs.trendmicro.com/en-us/tippingpoint/threat-protection-system.aspx.

- Trend Micro TippingPoint Threat Protection System Hardware Specification and Installation Guide, September 2020
- Trend Micro TippingPoint Threat Protection System Command Line Interface Reference, July 2021
- Trend Micro TippingPoint Virtual Threat Protection System (vTPS) User Guide, October 2021
- Trend Micro TippingPoint Threat Protection System Common Criteria Evaluated Configuration Guide (CCECG) for TPS v5.5.0, Document Version 1.0, 22 May 2023

3. Security Problem Definition

This section defines the security problem to be addressed by the TOE, in terms of threats to be countered by the TOE or its operational environment, assumptions about the intended operational environment of the TOE, and Organizational Security Policies (OSPs) that apply to the TOE.

3.1 Assumptions

This section contains assumptions regarding the operational environment and the intended usage of the TOE.

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.CONNECTIONS

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

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 computing platform for general purpose applications (unrelated to networking functionality).

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 and/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.

A.TRUSTED_ADMINISTRATOR

The authorized 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 being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The 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.

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.

3.2 Threats

T.NETWORK_ACCESS

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_DISCLOSURE

Sensitive information on a protected network might be disclosed to an attacker resulting from ingress- or egress-based actions.

T.NETWORK_DOS

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.

T.NETWORK_MISUSE

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, forced resets, malicious zip files, disguised executables, privilege escalation tools, and botnets).

T.PASSWORD_CRACKING

Malicious users or external IT entities 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_COMPROMISE

Malicious users or external IT entities 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.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. UNAUTHORIZED_ADMINISTRATOR_ACCESS

A user 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.UNDETECTED_ACTIVITY

Users or external IT entities 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.UNTRUSTED_COMMUNICATION_CHANNELS

Malicious remote users or external IT entities may attempt to target network devices that do not use standardized secure tunneling 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 could 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. UPDATE_COMPROMISE

Users 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.WEAK_AUTHENTICATION_ENDPOINTS

Malicious remote users or external IT entities may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., shared password that is guessable or transported as plaintext. Successful attacks could allow the attacker to 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.

T. WEAK_CRYPTOGRAPHY

Malicious remote users or external IT entities may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Weak encryption algorithms, modes, or inappropriate key sizes could 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.

3.3 Organizational Security Policies

This section describes the Organizational Security Policies (OSPs) that apply to the TOE. OSPs are used to provide a basis for security objectives that are commonly desired by TOE Owners in this operational environment, but for which it is not practical to universally define the assets being protected or the threats to those assets.

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

Analytical processes and information to derive conclusions about potential intrusions must be applied to IPS data and appropriate response actions taken.

4. Security Objectives

This section identifies the security objectives for the TOE and its operational environment. The security objectives identify the responsibilities of the TOE and its environment in addressing the security problem defined in Section 3.

4.1 Security Objectives for the Operational Environment

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.CONNECTIONS

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.

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.

OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.TRUSTED_ADMIN

Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

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.

4.2 Security Objectives for the TOE

O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security relevant events associated with users; and store those audit data locally, or externally if configured.

O.DISPLAY_BANNER

The TOE will display an advisory warning regarding use of the TOE.

O.IPS_ANALYZE

Entities that reside on or communicate across monitored networks must have network activity effectively analyzed for potential violations of approved network usage. The TOE must be able to effectively analyze data collected from monitored networks to reduce the risk of unauthorized disclosure of information, inappropriate access to services, and misuse of network resources.

O.IPS_REACT

The TOE must be able to react in real-time as configured by the Security Administrator to terminate and block traffic flows that have been determined to violate administrator-defined IPS policies.

O. PROTECTED_COMMUNICATIONS

The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.

O.SECURITY_FUNCTIONALITY_COMPROMISE
The TOE will properly destroy cryptographic keys when they are no longer needed and will prevent the reading of pre-shared keys, symmetric keys, and private keys.

O.STRONG_CRYPTOGRAPHY
The TOE will provide strong standards-based cryptographic algorithms and key sizes.

O.SYSTEM_MONITORING
To be able to analyze and react to potential network policy violations, the IPS must be able to collect and store essential data elements of network traffic on monitored networks.

O.TOE_ADMINISTRATION
The TOE will provide mechanisms to ensure that only administrators are able to log in and use the management interfaces to configure the TOE and its network, mechanisms that control a user's logical access to the TOE and mechanisms to ensure strong passwords.

O.TSF_SELF_TEST
The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

O.VERIFIABLE_UPDATES
The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and cryptographically validated to be from a trusted source.

5. IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The security requirements for the TOE have been drawn from Parts 2 and 3 of the Common Criteria. The security functional requirements have been selected to correspond to the actual security functions implemented by the TOE while the assurance requirements have been selected to offer a low to moderate degree of assurance that those security functions are properly realized.

5.1 Extended Requirements

This Security Target includes Security Functional Requirements (SFRs) that are not drawn from CC Part 2. These Extended SFRs are identified by having a label '_EXT' after the requirement name for TOE SFRs. The structure of the extended SFRs is modeled after the SFRs included in CC Part 2. The structure is as follows:

A. Class – The extended SFRs included in this ST are part of the identified classes of requirements.

B. Family – The extended SFRs included in this ST are part of several SFR families including the new families defined below.

C. Component – The extended SFRs are not hierarchical to any other components, though they may have identifiers terminating in other than "1". The dependencies for the extended components are identified both in this section and in the TOE SFR Dependencies section of this ST (Section 7.3, Requirement Dependency Rationale).

5.1.1 Extended Family Definitions

5.1.1.1 Class FAU: Security Audit

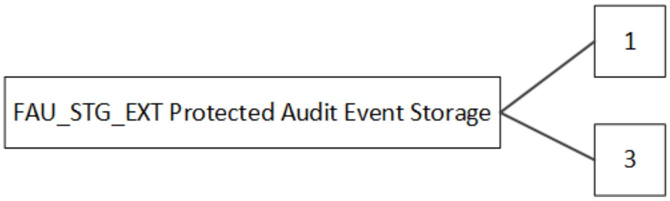
This class is defined as specified in the CC Part 2 for a TOE's ability to provide security auditing functionality.

FAU_STG_EXT

Family Behavior

This family requires that the TOE provide the ability to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC, and to store audit data locally. The .3 element requires the TSF to generate a warning to inform the Administrator before the audit trail exceeds the local audit trail storage capacity.

Component Levelling



Management: FAU_STG_EXT.1, FAU_STG_EXT.3

The following actions could be considered for the management functions in FMT:

a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_STG_EXT.1, FAU_STG_EXT.3

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) No audit necessary.

FAU_STG_EXT.1 – Protected audit event storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FTP_ITC.1 Inter-TSF Trusted Channel

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.

FAU_STG_EXT.3 – Action in Case of Possible Audit Data Loss

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.3.1 The TSF shall generate a warning to inform the Administrator before the audit trail exceeds the local audit trail storage capacity.

5.1.1.2 Class FCS: Cryptographic Support

This class is defined as specified in the CC Part 2 for a TOE’s ability to employ cryptographic functionality.

FCS_RBG_EXT

Family Behavior

Components in this family address the requirements for random bit/number generation. This is a new family defined for the FCS class.

Component Levelling

FCS_RBG_EXT Random Bit Generation
1

Management: FCS_RBG_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_RBG_EXT.1

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) No audit necessary.

FCS_RBG_EXT.1– Random Bit Generation

Hierarchical to: No other components.

Dependencies: No other components.

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

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***selection: [assignment: number of software-based sources***] software-based noise source, [**assignment: number of platform-based sources] platform-based noise source**] with a minimum of [***selection: 128 bits, 192 bits, 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.

FCS_SHC_EXT

Family Behavior

Components in this family address the ability for a client to use SSH to protect data between the client and a server using the SSH protocol. This is a new family defined for the FCS class.

Component Levelling

Management: FCS_SHC_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_SHC_EXT.1

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) Failure of SSH session establishment

b) SSH session establishment

c) SSH session termination.

FCS_SHC_EXT.1 – SSH Client Protocol

Hierarchical to: No other components.

Dependencies: FCS_CKM.1Cryptographic Key Generation

FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)

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

FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)

FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)

FCS_RBG_EXT.1 Random Bit Generation

FCS_SHC_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [***selection: 4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308 section 3.1, 8332***].

FCS_SHC_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [***selection: password-based, no other method***].

FCS_SHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [**assignment: number of bytes**] bytes in an SSH transport connection are dropped.

FCS_SHC_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [**assignment: list of encryption algorithms**].

FCS_SHC_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [***selection: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa,***

ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [**assignment: list of data integrity MAC algorithms**] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SHC_EXT.1.7 The TSF shall ensure that [**assignment: list of key exchange method**s] are the only allowed key exchange methods used for the SSH protocol.

FCS_SHC_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_SHC_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 [***selection: a list of trusted certification authorities, no other methods***] as described in RFC 4251 section 4.1.

FCS_SHS_EXT

Family Behavior

Components in this family address the ability for a server to offer SSH to protect data between a client and the server using the SSH protocol. This is a new family defined for the FCS class.

Component Levelling

Management: FCS_SHS_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_SHS_EXT.1

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) Failure of SSH session establishment

b) SSH session establishment

c) SSH session termination.

FCS_SHS_EXT.1 – SSH Server Protocol

Hierarchical to: No other components.

Dependencies: FCS_CKM.1Cryptographic Key Generation

FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)

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

FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)

FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)

FCS_RBG_EXT.1 Random Bit Generation

FCS_SHS_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [***selection: 4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308 section 3.1, 8332***].

FCS_SHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [***selection: password-based, no other method***].

FCS_SHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [**assignment: number of bytes**] bytes in an SSH transport connection are dropped.

FCS_SHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [**assignment: list of encryption algorithms**].

FCS_SHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [***selection: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256***] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [**assignment: list of data integrity MAC algorithms**] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SHS_EXT.1.7 The TSF shall ensure that [**assignment: list of key exchange methods**] are the only allowed key exchange methods used for the SSH protocol.

FCS_SHS_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.

5.1.1.3 Class FIA: Identification and Authentication

This class is defined as specified in the CC Part 2 for a TOE's ability to provide functionality to establish and verify a claimed user identity.

FIA_PMG_EXT

Family Behavior

Components in this family address the requirement to define the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

Component Levelling

Management: FIA_PMG_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FIA_PMG_EXT.1

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) No specific audit requirements

FIA_PMG_EXT.1 – Password Management

Hierarchical to: No other components.

Dependencies: No other components.

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: [***selection: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [assignment: other characters***]];

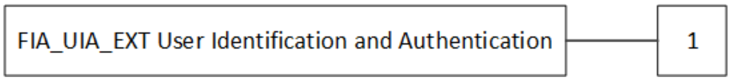
b) Minimum password length shall be configurable to between [**assignment: minimum number of characters supported by the TOE**] and [**assignment: number of characters greater than or equal to 15**] characters.

FIA_UIA_EXT

Family Behavior

Components in this family specifies certain actions that the TOE allows before the non-TOE entity goes through the required identification and authentication process.

Component Levelling



Management: FIA_UIA_EXT.1

The following actions could be considered for the management functions in FMT:

a) Ability to configure the list of TOE services available before an entity is identified and authenticated.

Audit: FIA_UIA_EXT.1

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) All use of the identification and authentication mechanism.

FIA_UIA_EXT.1 – User Identification and Authentication

Hierarchical to: No other components.

Dependencies: FTA_TAB.1 Default TOE Access Banners.

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;
- [***selection: no other actions, automated generation of cryptographic keys, [assignment: list of services, actions performed by the TSF in response to non-TOE requests]***].

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.

FIA_UAU_EXT

Family Behavior

Components in this family provides for a locally based administrative user authentication mechanism.

Component Levelling

Management: FIA_UAU_EXT.2

The following actions could be considered for the management functions in FMT:

a) None.

Audit: FIA_UAU_EXT.2

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) Minimal: All use of the authentication mechanism.

FIA_UAU_EXT.2 – Password-based Authentication Mechanism

Hierarchical to: No other components.

Dependencies: No other components.

FIA_UAU_EXT.2.1 The TSF shall provide a local [***selection: password-based, SSH public key-based, certificate-based, [assignment: other authentication mechanism(s)]***] authentication mechanism to perform local administrative user authentication.

5.1.1.4 Class FPT: Protection of the TSF

This class is defined as specified in the CC Part 2 providing functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data.

Each new family defined is modelled after the FPT_PTD Class.

FPT_APW_EXT

Family Behavior

Components in this family address the requirements for protecting plaintext credential data such as passwords from unauthorized disclosure.

Component Levelling

FPT_APW_EXT Protection of Administrator Passwords
1

Management: FPT_APW_EXT.1

The following actions could be considered for the management functions in FMT:

a) No management functions.

Audit: FPT_APW_EXT.1

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) No audit necessary.

FPT_APW_EXT.1 – Protection of Administrator Passwords

FPT_APW_EXT.1 Protection of Administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.

Hierarchical to: No other components.

Dependencies: No other components.

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

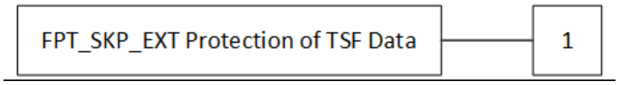
FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext administrative passwords.

FPT_SKP_EXT

Family Behavior

Components in this family address the requirements for protecting TSF data, such as cryptographic keys.

Component Levelling



Management: FPT_SKP_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FPT_SKP _EXT.1

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) Minimal: All use of the authentication mechanism.

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

FPT_SKP_EXT.1 Protection of TSF Data (for Reading of all Pre-shared, Symmetric and Private Keys), requires preventing pre-shared, symmetric and private keys from being read by any user or subject. It is the only component of this family.

Hierarchical to: No other components.

Dependencies: No other components.

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

FPT_STM_EXT

Family Behavior

Components in this family extend FPT_STM requirements by describing the source of time used in timestamps.

Component Levelling

Management: FPT_STM_EXT.1

The following actions could be considered for the management functions in FMT:

a) Management of the time

b) Administrator setting of the time.

Audit: FPT_STM_EXT.1

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) Discontinuous changes to the time.

FPT_STM_EXT.1 – Reliable Time Stamps

FPT_STM_EXT.1 Reliable Time Stamps is hierarchic to FPT_STM.1: it requires that the TSF provide reliable time stamps for TSF and identifies the source of the time used in those timestamps.

Hierarchical to: No other components.

Dependencies: No other components.

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

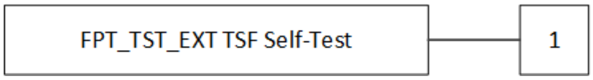
FPT_STM_EXT.1.2 The TSF shall [***selection: allow the Security Administrator to set the time, synchronize time with an NTP server***].

FPT_TST_EXT

Family Behavior

Components in this family address the requirements for self-testing the TSF for selected correct operation.

Component Levelling



Management: FPT_TST_EXT.1

The following actions could be considered for the management functions in FMT:

a) Ability to set the time which is used for time-stamps.

Audit: FPT_TST_EXT.1

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) Indication that TSF self-test was completed

b) Failure of self-test.

FPT_TST_EXT.1 – TSF Testing

FPT_TST_EXT.1 TSF Self-Test requires a suite of self-tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

Hierarchical to: No other components.

Dependencies: No other components.

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [***selection: during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]***] to demonstrate the correct operation of the TSF: [**assignment: list of self-tests run by the TSF**].

FPT_TUD_EXT

Family Behavior

Components in this family address the requirements for updating the TOE firmware and/or software.

Component Levelling

Management: FPT_TUD_EXT.1

The following actions could be considered for the management functions in FMT:

a) Ability to perform a manual TOE update.
b) Ability to verify the updates.

Audit: FPT_TUD_EXT.1

The following actions should be considered for audit if FAU_GEN.1 Security audit data generation is included:

a) Any attempt to initiate a manual update.

FPT_TUD_EXT.1 – Trusted Update

FPT_TUD_EXT.1 Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

Hierarchical to: No other components.

Dependencies: FCS_COP.1/SigGen Cryptographic operation (for Cryptographic Signature and Verification), or

FCS_COP.1/Hash Cryptographic operation (for cryptographic hashing).

FPT_TUD_EXT.1.1 The TSF shall provide [**assignment: Administrators**] the ability to query the currently executing version of the TOE firmware/software and [***selection: the most recently installed version of the TOE firmware/software; no other TOE firmware/software version***].

FPT_TUD_EXT.1.2 The TSF shall provide [**assignment: Administrators**] the ability to manually initiate updates to TOE firmware/software and [***selection: support automatic checking for updates, support automatic updates, no other update mechanism***].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [***selection: X.509 certificate, digital signature, published hash***] prior to installing those updates.

5.1.1.5 Class IPS: Intrusion Prevention System

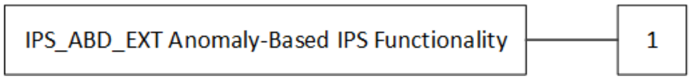
This class is defined specifically for Intrusion Prevention security functionality that is not defined in CC Part 2. Intrusion prevention involves the TOE's ability to collect network packets, examine their contents for information that suggests malicious activity, and to perform some action in response such as terminating the connection.

IPS_ABD_EXT Anomaly-Based IPS Functionality

Family Behavior

This family defines requirements for detection of anomalous network traffic and how the TSF should respond if an anomaly is detected.

Component Levelling



Management: IPS_ABD_EXT.1

The following actions could be considered for the management functions in FMT:

a) Configuration of anomaly detection.

b) Enabling and disabling actions to be taken when anomaly matches are detected.

c) Modification of thresholds that trigger IPS reactions.

d) Modification of the duration of traffic blocking actions.

Audit: IPS_ABD_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST: a) Inspected traffic that matches an anomaly-based IPS policy.

IPS_ABD_EXT.1 – Anomaly-Based IPS Functionality

IPS_ABD_EXT.1 Anomaly-Based IPS Functionality, requires the TSF to detect anomalous network traffic based on some criteria and to define the response that is issued if an anomaly is detected.

Hierarchical to: No other components.

Dependencies: IPS_NTA_EXT.1 Network Traffic Analysis

IPS_SBD_EXT.1 Signature-Based IPS Functionality

IPS_ABD_EXT.1.1 The TSF shall support the definition of [***selection: baselines ('expected and approved'), anomaly ('unexpected') traffic patterns***] including the specification of [***assignment: attributes or characteristics of network traffic.***]

IPS_ABD_EXT.1.2 The TSF shall support the definition of anomaly activity through [***selection: 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: [***assignment: action taken by TSF in response to detection of anomaly***]
- In inline mode: [***assignment: action taken by TSF In response to detection of anomaly***].

IPS_IPB_EXT IP Blocking

Family Behavior

This family defines requirements for handling of inspected network traffic based on IP address.

Component Levelling

Management: IPS_IPB_EXT.1

The following actions could be considered for the management functions in FMT:

a) Modification of the known-good and known-bad lists (of IP addresses or address ranges).

b) Configuration of the known-good and known-bad lists to override signature-based IPS policies.

Audit: IPS_IPB_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST: a) Inspected traffic matches a list of known-good or known-bad addresses applied to an IPS policy.

IPS_IPB_EXT.1 – IP Blocking

IPS_IPB_EXT.1 IP Blocking, requires the TSF to enforce IPS policies that are based on IP address.

Hierarchical to: No other components.

Dependencies: IPS_NTA_EXT.1 Network Traffic Analysis

FMT_SMR.1 Security Roles

IPS_IPB_EXT.1.1 The TSF shall support configuration and implementation of known-good and known-bad lists of [***selection: source, destination***] IP addresses and [***selection: no additional address types, [assignment: list of address types]***].

IPS_IPB_EXT.1.2 The TSF shall allow [assignment: authorized roles] to configure the following IPS policy elements: [***assignment: IPS policy elements***].

IPS_NTA_EXT Network Traffic Analysis

Family Behavior

This family defines the network traffic protocols the TOE is capable of analyzing and detecting violations for.

Component Levelling

Management: IPS_NTA_EXT.1

The following actions could be considered for the management functions in FMT:

a) Modification of the parameters that define the network traffic to be collected and analyzed.

Audit: IPS_NTA_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Modification of which IPS policies are active on a TOE interface.

b) Enabling/disabling a TOE interface with IPS policies applied.

c) Modification of which mode(s) is/are active on a TOE interface.

IPS_NTA_EXT.1 – Network Traffic Analysis

IPS_NTA_EXT.1 Network Traffic Analysis, requires the TSF to be able to inspect traffic for certain network protocols and in certain architectural deployments.

Hierarchical to: No other components.

Dependencies: No dependencies.

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:

- [***assignment: network protocols and any standard(s) that define their implementation***].

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: [***assignment: list of interface types***];
- Inline (data pass-through) mode: [***assignment: list of interface types***];
- Management mode: [***assignment: list of interface types***];
- [***selection:***
    - ***o [assignment: other interface types];***
    - ***o no other interface types***].

IPS_SBD_EXT Signature-Based IPS Functionality

Family Behavior

This family defines requirements for analysis of network traffic based on packet characteristics.

Component Levelling

Management: IPS_SBD_EXT.1

The following actions could be considered for the management functions in FMT:

a) Enabling and disabling signatures applied to sensor interfaces.

b) Updating (importing) signatures.

c) Creating custom signatures.

d) Enabling and disabling actions to be taken when signature matches are detected.

Audit: IPS_SBD_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Inspected traffic matches a signature-based IPS rule with logging enabled.

IPS_SBD_EXT.1 – Signature-Based IPS Functionality

IPS_SBD_EXT.1 Signature-Based IPS Functionality, requires the TSF to detect network traffic with certain packet characteristics and take some action when this traffic is detected.

Hierarchical to: No other components.

Dependencies: IPS_NTA_EXT.1 Network Traffic Analysis

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: [***assignment: applicable header fields for each supported network protocol***].

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: [***assignment: applicable packet payload data elements for each supported network protocol***].

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: [***assignment: applicable header-based signatures for identified header fields***].

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: [***assignment: list of traffic patterns***].

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: **[*assignment: action taken by TSF in response to detection of signature***]

• In inline mode:

o block/drop the traffic flow;

o and [***assignment: action taken by TSF in response to detection of signature***].

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.

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the Trend Micro TPS TOE.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1/Audit: Audit Data Generation (Audit)
	FAU_GEN.1/IPS: Audit Data Generation (IPS)
	FAU_GEN.2: User Identity Association
	FAU_STG.1/Audit: Protected Audit Trail Storage (Audit Data)
	FAU_STG.1/IPS: Protected Audit Trail Storage (IPS Data)
	FAU_STG_EXT.1: Protected Audit Event Storage
	FAU_STG_EXT.3: Action in Case of Possible Audit Data Loss
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation
	FCS_CKM.2: Cryptographic Key Distribution

Requirement Class	Requirement Component
	FCS_CKM.4: Cryptographic Key Destruction
	FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/ Decryption)
	FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1/Hash : Cryptographic Operation (Hash Algorithm)
	FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)
	FCS_RBG_EXT.1: Random Bit Generation
	FCS_SHC_EXT.1: SSH Client Protocol
	FCS_SHS_EXT.1: SSH Server Protocol
FIA: Identification and authentication	FIA_AFL.1: Authentication Failure Handling
	FIA_PMG_EXT.1: Password Management
	FIA_UIA_EXT.1: User Identification and Authentication
	FIA_UAU_EXT.2: Password-based Authentication Mechanism
	FIA_UAU.7: Protected Authentication Feedback
FMT: Security Management	FMT_MOF.1/ManualUpdate: Management of Security Functions Behaviour
	FMT_MOF.1/Functions: Management of Security Functions Behaviour
	FMT_MTD.1: Management of TSF Data
	FMT_SMF.1/Core: Specification of Management Functions (Core)
	FMT_SMF.1/IPS Specification of Management Functions (IPS)
	FMT_SMR.2: Restrictions on Security Roles
FPT: Protection of the TSF	FPT_APW_EXT.1: Protection of Administrator Passwords
	FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	FPT_STM_EXT.1: Reliable Time Stamps
	FPT_TST_EXT.1: TSF Testing
	FPT_TUD_EXT.1: Trusted Update
FTA: TOE access	FTA_SSL.3: TSF-initiated Termination
	FTA_SSL.4: User-initiated Termination
	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF Trusted Channel
	FTP_TRP.1: Trusted Path
IPS: Intrusion Prevention	IPS_ABD_EXT.1: Anomaly-Based IPS Functionality
	IPS_IPB_EXT.1: IP Blocking
	IPS_NTA_EXT.1: Network Traffic Analysis
	IPS_SBD_EXT.1: Signature-Based IPS Functionality

Table 3 TOE Security Functional Components

5.2.1 Security audit (FAU)

5.2.1.1 Audit Data Generation (Audit) (FAU_GEN.1/Audit)

FAU_GEN.1.1/Audit 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) All auditable events for the [**not specified**] level of audit; and

c) [

All administrative actions comprising:

- **Administrative login and logout.**
- **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); and**
- **Specifically defined auditable events listed in Table 4**].

FAU_GEN.1.2/Audit 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 (if applicable), 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 ~~PP/~~ST, [**information specified in column three of Table 4**].

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1/Audit	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1/Audit	None.	None.
FAU_STG.1/IPS	None.	None.
FAU_STG_EXT.1	None.	None.
FAU_STG_EXT.3	Low storage space for audit events.	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.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_SHC_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_SHS_EXT.1	Failure to establish an SSH session	Reason for failure
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 identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None

Requirement	Auditable Events	Additional Audit Record Contents
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MTD.1	None.	None.
FMT_SMF.1/Core	All management activities of the TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
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.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FTA_SSL.3	The termination of a local or remote interactive session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an administrator's own interactive session by the administrator.	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	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions	None.

Table 4 Auditable Events

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

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

a) Start-up and shut-down of the **IPS** audit 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; and**
f) **Totals of similar reactions occurring within a specified time period**].

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

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

b) For each **IPS** audit record, based on the auditable event definitions of the functional components included in the ~~PP/~~ST, [**Specifically defined auditable events listed in Table 5**].

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 of 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).

Table 5 IPS Auditable Events

5.2.1.3 User Identity Association (FAU_GEN.2)

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

FAU_STG.1.1/Audit The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2/Audit The TSF shall be able to [***prevent***] unauthorised modifications to the stored audit records in the audit trail.

Application Note: *The TOE protects both IPS data and the data in other audit records. This iteration refers to the non-IPS audit records.*

5.2.1.5 Protected Audit Trail Storage (IPS Data) (FAU_STG.1/IPS)

FAU_STG.1.1/IPS The TSF shall protect the stored **IPS** audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2/IPS The TSF shall be able to [***prevent***] unauthorised modifications to the stored **IPS** audit records in the audit trail.

Application Note: *The TOE protects both IPS data and the data in other audit records. This iteration refers to the audit records containing IPS data.*

5.2.1.6 Action in Case of Possible Audit Data Loss (FAU_STG_EXT.3)

FAU_STG_EXT.3.1 The TSF shall generate a warning to inform the Administrator before the audit trail exceeds the local audit trail storage capacity.

5.2.1.7 Protected Audit Event Storage (FAU_STG_EXT.1)

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 consist of a single standalone component that stores audit data locally.***]

FAU_STG_EXT.1.3 The TSF shall [***overwrite previous audit records according to the following rule: [the oldest historical audit file is deleted, the current audit file is renamed as a historical audit file, and a new current audit file is created]***] when the local storage space for audit data is full.

5.2.2 Cryptographic support (FCS)

5.2.2.1 Cryptographic Key Generation (FCS_CKM.1)

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, P-521 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;**
- **FFC Schemes using "safe-prime" groups with cryptographic key sizes 2048-bits, 3072-bits, and 4096-bits that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key**

Establishment Schemes Using Discrete Logarithm Cryptography" and RFC 3526]

and specified cryptographic key sizes **or NIST P-curves [cryptographic key sizes and NIST P-curves as identified in the assignments above]** that meet the following: **[standards as identified in the assignments above].**

5.2.2.2 Cryptographic Key Distribution (FCS_CKM.2)

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

- **Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";**
- **FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and groups listed in RFC 3526**.]

that meets the following: [**standards as identified in the assignments above**].

5.2.2.3 Cryptographic Key Destruction (FCS_CKM.4)

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 a new value of the key.**

]
that meets the following: [**key zeroization (i.e. cryptographic key destruction) as defined in FIPS 140-2**].

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

FCS_COP.1.1/DataEncryption The TSF shall perform [**encryption/decryption**] in accordance with a specified cryptographic algorithm [**AES used in CBC, GCM] mode** and cryptographic key sizes [**128 bits, 256 bits**] that meet the following: [**AES as specified in ISO 18033-3**, **CBC as specified in ISO 10116, GCM as specified in ISO 19772**].

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

FCS_COP.1.1/SigGen The TSF shall perform [**cryptographic signature services**: **generation and verification**] in accordance with a specified cryptographic algorithm **and cryptographic key sizes** [

- **RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits],**
- **Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits, 384-bits, 521-bits]]** ~~**and cryptographic key sizes [assignment: cryptographic key sizes]**~~

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" [selection: P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4**

].

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

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~~ **message digest sizes** [**160, 256, 384, 512 bits**] that meet the following: [**ISO/IEC 10118-3:2004**].

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

FCS_COP.1.1/KeyedHash The TSF shall perform [**keyed-hash message authentication**] in accordance with a specified cryptographic algorithm [**HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512, implicit**] and cryptographic key sizes [**160, 256, 512 bits**] **and message digest sizes** [**160, 256, 512 bits**] that meet the following: [**ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"**].

5.2.2.8 Random Bit Generation (FCS_RBG_EXT.1)

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)***].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[two] platform-based noise sources***] 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.

5.2.2.9 SSH Client Protocol (FCS_SHC_EXT.1)

FCS_SHC_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFC(s) 4251, 4252, 4253, 4254, [***5647, 5656, 6668***].

FCS_SHC_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [***no other method***].

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

FCS_SHC_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa***, ***ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521***] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [**hmac-sha1**, **hmac-sha2-256**, **hmac-sha2-512**, **implicit**] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SHC_EXT.1.7 The TSF shall ensure that [**diffie-hellman-group14-sha1**, **ecdh-sha2-nistp256**] and [**ecdh-sha2-nistp384, ecdh-sha2-nistp521**] are the only allowed key exchange methods used for the SSH protocol.

FCS_SHC_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_SHC_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.

5.2.2.10 SSH Server Protocol (FCS_SHS_EXT.1)

FCS_SHS_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [***5647, 5656, 6668, 8268***].

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

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

FCS_SHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521***] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [**hmac-sha1, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM**] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SHS_EXT.1.7 The TSF shall ensure that [**diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, diffie-hellman-group16-sha512**] are the only allowed key exchange methods used for the SSH protocol.

FCS_SHS_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 Identification and authentication (FIA)

5.2.3.1 Authentication Failure Handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when an [***Administrator configurable positive integer within [1 to 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 remote Administrator from successfully authenticating until an Administrator defined time period has elapsed**].

5.2.3.2 Password Management (FIA_PMG_EXT.1)

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: ***[“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [“,”, “.”, “/”, “<”, “>”, “?”]***];

b) Minimum password length shall be configurable to between [**1**] and [**15**] characters**.**

Application Note: *The minimum password length can be configured to "1", "8", or "15" only*.

5.2.3.3 Protected Authentication Feedback (FIA_UAU.7)

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**.

5.2.3.4 Password-based Authentication Mechanism (FIA_UAU_EXT.2)

FIA_UAU_EXT.2.1 The TSF shall provide a local [***password-based, SSH public key-based***] authentication mechanism to perform local administrative user authentication.

5.2.3.5 User Identification and Authentication (FIA_UIA_EXT.1)

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;
- **[*[send Echo Reply in response to Echo Request ICMP messages received at the Management interface]*]**.

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 Security management (FMT)

5.2.4.1 Management of Security Functions Behaviour (FMT_MOF.1/ManualUpdate)

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

5.2.4.2 Management of Security Functions Behaviour (FMT_MOF.1/Functions)

FMT_MOF.1.1/Functions The TSF shall restrict the ability to [***determine the behaviour of; modify the behaviour of***] the functions [**transmission of audit data to an external IT entity**] to [**Security Administrators**].

5.2.4.3 Management of TSF Data (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to [***[manage]***] the [**TSF data**] to [**Security Administrators**].

5.2.4.4 Specification of Management Functions (FMT_SMF.1/Core)

FMT_SMF.1.1/Core 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 local log file size; clear local log files);**
- **Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;**
- **Ability to configure the cryptographic functionality;**

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

].

5.2.4.5 Specification of Management Functions (IPS) (FMT_SMF.1/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)**
- **Update (import) 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**].

5.2.4.6 Restrictions on Security Roles (FMT_SMR.2)

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.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 Protection of Administrator Passwords (FPT_APW_EXT.1)

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.

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

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

5.2.5.3 Reliable Time Stamps (FPT_STM_EXT.1)

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, synchronize time with an NTP server*].

5.2.5.4 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [

- **software module integrity tests**
- **cryptographic known answer tests**

].

5.2.5.5 Trusted Update (FPT_TUD_EXT.1)

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.

5.2.6 TOE access (FTA)

5.2.6.1 TSF-initiated Termination (FTA_SSL.3)

FTA_SSL.3.1 The TSF shall terminate **local and remote** interactive session**(s)** after a**n** [**inactivity time interval from 1 to 32000**].

5.2.6.2 User-initiated Termination (FTA_SSL.4)

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

5.2.6.3 Default TOE Access Banners (FTA_TAB.1)

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

5.2.7 Trusted path/channels (FTP)

5.2.7.1 Inter-TSF Trusted Channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall provide a **trusted** communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit [***the TSF***] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [**transmitting audit records to an external audit server**].

5.2.7.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1 The TSF provide a communication path between itself and [**remote**] users 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 The TSF shall permit [***remote users*]** to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [***initial user authentication, [all remote administrative actions]***].

5.2.8 Intrusion Prevention System (IPS)

5.2.8.1 Anomaly-Based IPS Functionality (IPS_ABD_EXT.1)

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

- ***throughput (kilobytes per second);***

].

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

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, destination, port] unreachable message**]
- In inline mode:
  - [**allow the traffic flow**
  - **block/drop the traffic flow**].

5.2.8.2 IP Blocking (IPS_IPB_EXT.1)

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**].

5.2.8.3 Network Traffic Analysis (IPS_NTA_EXT.1)

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 (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: [**none**];
- Inline (data pass-through) mode: [
  - **1 GBE,**
  - **10 GBE,**
  - **40 GB Ethernet,**
  - **virtio for KVM hypervisor, and**

  - ○ **vmxnet3 for VMware's hypervisor**];
- Management mode: [**1 Gb Ethernet and serial**];
- [***no other interface types***].

5.2.8.4 Signature-Based IPS Functionality (IPS_SBD_EXT.1)

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; home address options, traffic class, flow label.**
- **ICMP: Type; Code; Header Checksum; and [*ID, sequence number*].**
- **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) **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.**
- **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;**]
- In inline mode:
  - block/drop the traffic flow;
  - and [
    - **allow all traffic flow**].

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 Security Assurance Requirements

The security assurance requirements for the TOE are the EAL components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.2: Security-enforcing functional specification
	ADV_TDS.1: Basic design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.2: Use of a CM system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1: Delivery procedures
ASE: Security Target	ASE_CCL.1: Conformance claims
	ASE_ECD.1: Extended components definition
	ASE_INT.1: ST introduction
	ASE_OBJ.2: Security objectives
	ASE_REQ.2: Derived security requirements
	ASE_SPD.1: Security Problem Definition
	ASE_TSS.1: TOE summary specification
ATE: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1: Functional testing

Requirement Class	Requirement Component
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2: Vulnerability analysis

Table 6 Security Assurance Components

5.3.1 Development (ADV)

ADV_ARC.1 – Security architecture description

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

ADV_ARC.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2 – Security-enforcing functional specification

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.2D The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.2.1C The functional specification shall completely represent the TSF.

ADV_FSP.2.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4C For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.2.5C For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

ADV_TDS.1 – Basic design

ADV_TDS.1.1D The developer shall provide the design of the TOE.

ADV_TDS.1.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.1.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.1.2C The design shall identify all subsystems of the TSF.

ADV_TDS.1.3C The design shall describe the behaviour of each SFR-supporting or SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.

ADV_TDS.1.4C The design shall summarise the SFR-enforcing behaviour of the SFR-enforcing subsystems.

ADV_TDS.1.5C The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.

ADV_TDS.1.6C The mapping shall demonstrate that all TSFIs trace to the behaviour described in the TOE design that they invoke.

ADV_TDS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.1.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance Documents (AGD)

AGD_OPE.1 – Operational user guidance

AGD_OPE.1.1D The developer shall provide operational user guidance.

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1 – Preparative procedures

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.3.3 Life-cycle (ALC)

ALC_CMC.2 – Use of a CM system

ALC_CMC.2.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.2.2D The developer shall provide the CM documentation.

ALC_CMC.2.3D The developer shall use a CM system.

ALC_CMC.2.1C The TOE shall be labelled with its unique reference.

ALC_CMC.2.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.2.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_CMS.2 – Parts of the TOE CM coverage

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

ALC_CMS.2.1C The configuration list shall include the following: The TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.

ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.2.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

ALC_CMS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DEL.1 – Delivery procedures

ALC_DEL.1.1D The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

ALC_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Security Target Evaluation (ASE)

ASE_CCL.1 – Conformance claims

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

ASE_CCL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1 – Extended components definition

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

ASE_ECD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

ASE_INT.1 – ST introduction

ASE_INT.1.1D The developer shall provide an ST introduction.

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

ASE_INT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

ASE_OBJ.2 – Security objectives

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

ASE_OBJ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_REQ.2 – Derived security requirements

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

ASE_REQ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_SPD.1 – Security problem definition

ASE_SPD.1.1D The developer shall provide a security problem definition.

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

ASE_SPD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1 – TOE summary specification

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

ASE_TSS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.3.5 Tests (ATE)

ATE_COV.1 – Evidence of coverage

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 – Functional testing

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 – Independent testing - sample

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.3.6 Vulnerability Assessment (AVA)

AVA_VAN.2 – Vulnerability analysis

AVA_VAN.2.1D The developer shall provide the TOE for testing.

AVA_VAN.2.1C The TOE shall be suitable for testing.

AVA_VAN.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E	The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels
- Intrusion Prevention System

6.1 Security Audit

The TOE is a single standalone component that generates security relevant audit records including administrative activity. The audit records are stored locally on the TOE, protected from unauthorized modification and deletion and can be sent to a remote syslog server for storage. The connection for transmission of audit records uses SSH.

6.1.1 FAU_GEN.1/Audit: Audit Data Generation (Audit)

The TOE is able to generate audit records for security relevant events as they occur. The events that can cause an audit record to be logged include: starting and stopping the audit function; all attempts to initiate a secure communication channel; and any use of an administrator action via the CLI comprising:

- Administrative login and logout.
- 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, when a key is changed, it is uniquely identified in the audit log by being referenced as an SSH key and by identifying the username that the key is associated with).
- Resetting passwords (name of related user account is logged).
- Attempts to initiate a TOE update.
- Modification of the behaviour of the transmission of audit data to an external IT entity.

Additionally, the TOE generates an audit record warning that is written to the audit trail when the space allocated for storage of audit records exceeds 75% of capacity. This is default behavior and is not configurable.

The audit records include the following fields:

- Log ID - Displays the system-assigned log ID number.
- Log Entry Time - Displays the time the log was entered in the format YYYY-MM-DD HH:MM:SS.
- Device Name - The device name on which the session was logged.
- User - Displays the login name of the user who performed the audited action. The user listed for an event can include SYS and CLI.
- Access - Displays the access level of the user performing the action. This field is only present in Audit Log type.
- IP Address - Displays the IP address from which the user performed the action.
- Interface - Displays the interface with which the user logged in: CLI for the command line interface. For system-initiated actions, SYS displays in this field.
- Access - Displays the access level of the user performing the action (from 0 (no administrative permissions to 8 (super user)). In particular, the following values relevant to the TOE's evaluated configuration are defined for this field as follows:
    - 0 NORMAL_ACCESS (no administrative permissions)
    - 1 OPERATOR_ACCESS
    - 4 ADMINISTRATOR_ACCESS

    - 8 SUPER_USER_ACCESS
- Result - Displays the action performed or the result of a LOGIN or LOGOUT attempt.
- Action/Message – Text of the log entry identifying the action performed as a result. For example, Log Files Reset.

6.1.2 FAU_GEN.1/IPS: Audit Data Generation (IPS)

The TOE implements audit capabilities associated with IPS functions as specified in **Table 5** as well as the following events:

- Start-up and shut-down of the IPS functions;
- All IPS auditable events for the not specified level of audit;
- All dissimilar IPS events/reactions; and
- Totals of similar events/reactions occurring within a specified time period.

The TPS audit records include the date/time of the event, type of event and/or reaction, and additionally for each IPS auditable event type, based on the auditable event definitions of the functional components included in the ST, the specifically defined auditable events listed in **Table 5** column three.

Logging is enabled from within the Notification settings. Notification settings can be configured to log an event in response to a traffic-related event that occurs on the device. The traffic-related event can be the result of triggering an IPS filter configured with an action set that specifies a notification contact of Management Console and/or Remote System Log. Remote System logging sends messages to an external syslog server while Management Console stores the audit events and IPS data locally. These default contacts are available in all action sets and must be selected in order to store the audit events. The default aggregation time is 1 minute, configurable from 0 minutes to 10080 minutes using the aggregation period setting: Console >> Edit >> notifycontacts >> contact "Contact name" >> Period “value” from the CLI.

For a given type of event and depending on event types and event generation volume and frequency, the Aggregation capability creates a single event for similar events. The TOE uses alert aggregation to prevent system performance problems resulting from an excessive number of notification requests. Because a single packet can trigger an alert, attacks with large numbers of packets could potentially flood the alert mechanism used to send out notifications. The alert aggregation feature is used to trigger alert notifications at intervals to prevent this flooding. For example, when the aggregation interval is set to 5 minutes, the TOE sends an alert at the first IPS filter trigger, collects subsequent alerts, combines them into a single one and sends them out every five minutes. The aggregation period configured for a notification contact controls this alert aggregation.

For IPS_SBD_EXT.1, the TOE performs string-based pattern-matching on each field identified in Section 6.8.4 according to the signature based rules being applied to the traffic.

In the evaluated configuration, the TOE is configured to capture all header fields in the packet tracing associated with IPS logs.

6.1.3 FAU_GEN.2: User Identity Association

For audit events resulting from actions of identified users, the TOE associates each auditable event with the identity of the user that caused the event. Specifically, user identity is captured by the ‘User’ and ‘IP Address’ fields in the audit records.

6.1.4 FAU_STG.1/Audit: Protected Audit Trail Storage (Audit Data) / FAU_STG.1/IPS Protected Audit Trail Storage (IPS Data)

The TOE includes an internal log implementation that can be used to store audit records and IPS data locally on the TOE. The local audit logs are stored on the TOE hard drive in either the ‘Audit’ log or the ‘System’ log; and in the following logs for IPS data: “ipsAlert”, “ipsBlock”, “quarantine”, “reputationAlert”, and “reputationBlock”. The System log records information about the software processes that control the device, including startup and shutdown of the TOE events. All other required audit events as identified in **Table 4** are stored in the Audit log. The IPS data logs contain various data and alerts generated by the TOE's IPS functions and contain the required IPS audit events as identified in Table 5.

Each TPS system including vTPS systems allocate approximately one eighth of the system's internal disk space for the audit log file. Systems that support less than 5Gbps inspection throughput have 8GB internal disk space while systems that support 5Gbps and above inspection throughput have 32GB internal disk space. This implies approximately 1GB audit log file disk space on TPS systems with less than 5Gbps and approximately 4GB on TPS systems greater than 5Gbps.

The enforced limits on the size of the audit and IPS data logs are specified as a percentage of internal log disk space using the `log-file-size` CLI setting. The maximum amount of audit data that are stored locally in each log cannot exceed this percentage and the combined percentage configured for the logs must equal 100%.

The audit records and the IPS data files on the TOE are protected by database access control. There are no interfaces to modify or delete individual audit records. Only Super Users can configure the log sizes and clear the log files. The command: `clear log-file` deletes the locally stored Audit log and IPS data files. There are no interfaces to modify stored audit or IPS data.

6.1.5 FAU_STG_EXT.1: Protected Audit Event Storage

The TOE is capable of locally storing audit records and can be configured to send audit records to an external syslog server using SSH. When configured to send audit records to a syslog server, audit records are also written to the external syslog as they are written locally to the TOE audit log (in real-time). IPS logging can be configured to store the IPS data locally, remotely, or both.

Each TPS system including vTPS systems allocate approximately one eighth of the system's internal disk space for the audit log file. Systems that support less than 5Gbps inspection throughput have 8GB internal disk space while systems that support 5Gbps and above inspection throughput have 32GB internal disk space. This implies approximately 1GB audit log file disk space on TPS systems with less than 5Gbps and approximately 4GB on TPS systems greater than 5Gbps.

System disk space is monitored and once the available storage for audit trail exceeds 75% full an alert is generated. When audit storage space is exhausted, the TOE overwrites previous audit records by deleting the oldest historical log file, renaming the current log file to be a historical file, and creating a new current log file. There are 5 files by default for log rollover functionality (ex: audit.log -- current, audit.log.1..audit.log.4 -- rotated ones). Each file is allocated 20% of the total space allocated for that log. For example when the audit.log reaches its capacity (20% of audit log space) it is renamed to audit.log.1 and the new audit entries are written to audit.log. When audit.log reaches its capacity again, audit.log.1-->audit.log.2, audit.log-->audit.log.1 and new entries are written to audit.log. If all five files become filled (100% audit log space used) then the oldest file gets deleted.

6.1.6 FAU_STG_EXT.3: Action in Case of Possible Audit Data Loss

When the available storage for audit trail exceeds 75% full, the TOE generates an alert and writes it to the system log. This is default behavior and is not configurable.

6.2 Cryptographic Support

The TOE includes OpenSSL1.0.2l-fips wrapped with TippingPoint Crypto Core OpenSSL 2.0.13 library which provides cryptographic algorithms and services. The following functions have been certified in accordance with the identified standards. Note that two sets of algorithm certificates were awarded because the 1100TX and 5500TX were tested separately after their release.

Functions	Standards	Certificates
FCS_CKM.1 Cryptographic Key Generation		
• RSA (2048 bits)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	RSA #2945 Combined #C1262
• ECDSA (P-256, P-384, P-521 curves)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	ECDSA #1470

Functions	Standards	Certificates
		Combined #C1262
• FFC Schemes using ‘safe-prime’ groups (2048-bits, 3072-bits, 4096-bits)	NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and RFC 3526	N/A
FCS_CKM.2 Cryptographic Key Distribution		
• ECDSA (P-256, P-384, P-521 curves)	NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;	CVL #1937 Combined #C1262
• FFC Schemes using ‘safe-prime’ groups (2048-bits, 3072-bits, 4096-bits)	NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and RFC 3526	N/A
FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)		
• AES CBC (128 and 256 bits)	ISO 18033-3, CBC as specified in ISO 10116	AES #5484 Combined #C1262
• AES GCM (128 and 256 bits)	ISO 18033-3, GCM as specified in ISO 19772	AES #5484 Combined #C1262
FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)		
• RSA Digital Signature Algorithm (rDSA) (modulus 2048)	FIPS PUB 186-4 “Digital Signature Standard (DSS)”	RSA #2945 Combined #C1262
• ECDSA NIST curves (P-256, P-384, P-521)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves”; ISO/IEC 14888-3, Section 6.4	ECDSA #1470
FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)		
• SHA-1 (digest size 160 bits) • SHA-256 (digest size 256 bits) • SHA-384 (digest size 384 bits) • SHA-512 (digest size 512 bits)	ISO/IEC 10118-3:2004	SHS #4401 Combined #C1262
FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)		
• HMAC-SHA-1 (key size 160 bits and digest size 160 bits) • HMAC-SHA-256 (key size 256 bits and digest size 256 bits) • HMAC-SHA-512 (key size 512 bits and digest size 512 bits)	ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”	HMAC #3640 Combined #C1262
FCS_RBG_EXT.1: Random Bit Generation		
• CTR-DRBG(AES) with two independent platform-based noise	ISO/IEC 18031:2011	DRBG #2159 Combined #C1262

Functions	Standards	Certificates
source of 256 bits of non-determinism		

Table 7 Cryptographic Functions

6.2.1 FCS_CKM.1: Cryptographic Key Generation

The TOE generates RSA asymmetric keys using cryptographic key sizes of 2048 bits according to FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3. The RSA asymmetric keys are used in support of SSH public key authentication. See table above for Asymmetric key generation: RSA (2048-bit). The TOE generates asymmetric keys using FFC schemes using "safe-prime" groups in support of SSH key establishment. Diffie-Hellman groups 14, 15, and 16 are the supported key exchange methods and the key sizes are 2048 bits (Group 14), 3072 bits (Group 15), and 4096 bits (Group 16). The TOE also generates ECC asymmetric keys using NIST curves: P-256, P-384, P-521 in support of SSH public key authentication and SSH session establishment. Both key generation methods are used in communications with external syslog servers and with users accessing the SSH management interface.

6.2.2 FCS_CKM.2: Cryptographic Key Distribution

The TOE performs Cryptographic Key Distribution, or more specifically, key establishment using Diffie-Hellman group 14 that implements 2048-bit MODP Group according to RFC 3526, Section 3; Diffie-Hellman group 15 that implements 3072-bit MODP Group according to RFC 3526, Section 3; Diffie-Hellman group 16 that implements 4096-bit MODP Group according to RFC 3526, Section 3; and Elliptic Curve Diffie-Hellman key agreement using the P-256, P-384, or P-521 curve when an SSH cipher suite is negotiated. These key establishment methods are used during SSH session establishment with external audit server and with users accessing the SSH management interface.

See **Table 7 Cryptographic Functions** above for detail.

6.2.3 FCS_CKM.4: Cryptographic Key Destruction

The TOE uses the following secret keys, private keys and CSPs.

Key/CSP Name	Algorithm/Key Size	Description
RSA SGK	RSA 2048 bits	RSA signature generation key
RSA KDK	RSA 2048 bits	RSA key decryption key
ECC Keys	ECC key pair (P-256, P-384, P-521)	SSH session keys
SSH-RSA	RSA 2048 bits	SSH-RSA client keys
AES EDK	AES 128, 256 bits	AES encrypt/decrypt key
HMAC Key	HMAC-SHA-1 (160 bits) HMAC-SHA-256 (256 bits) HMAC-SHA-512 (512 bits)	HMAC keyed hash key
CTR_DRBG Key	AES 256 bits	Internal CTR_DRBG key variable

Table 8 Secret keys, Private keys and CSPs

The TOE incorporates OpenSSL, which provides implementation of the cryptographic algorithms specified in **Table 7**. The TOE operates in FIPS mode and invokes the OpenSSL crypto module APIs to set up and maintain the full SSH session, using the underlying cryptographic algorithms as identified in **Table 7**. Therefore, all key generation, negotiation of session keys, and packet authentication is performed and managed by the crypto module.

User passwords and SSH-RSA client keys are stored in internal flash, encrypted using AES with a 256 bit key encrypting key (KEK). When deleted, SSH_RSA client keys are overwritten with zeros. The KEK is stored on Compact Flash (CF) and is itself encrypted using AES with a 256 bit Master Key. For TPS appliances, the Master Key exists in hardware circuitry within the TOE. It is generated during manufacturing and is unique to each appliance.

For the vTPS, the Master Key is generated during software installation and stored on a system memory file. The vTPS implements multi-layer software obfuscation techniques (including masking and key wrapping) to protect the Master Key. These techniques protect the Master Key and associated authorization factors from unauthorized access. Anyone who has the copy of software or access to a running copy of software will not be able to access the plaintext Master Key by visually inspecting the software image, reverse engineering the software, or inspecting a memory footprint of a running software image. The Master Key and associated authorization factors are destroyed when the vTPS is factory reset (by execution of the debug factory-reset CLI command, or by deleting and reinstalling the VM). When this occurs, a new Master Key and authorization factors are generated, overwriting the previous values.

All remaining keys (see **Table 8** above) are plaintext stored in RAM, only during the lifetime of an API call. They are destroyed immediately after use, by overwriting the memory once with zeroes.

The TOE does not store plaintext keys in non-volatile memory and therefore this part of the requirement is trivially satisfied. Cryptographic Key Destruction is performed is accordance with key zeroization as described in FIPS 140-2.

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

The TOE performs 128/256-bit AES encryption/decryption as specified in ISO 18033-3, CBC mode as specified in ISO 10116 and GCM mode as specified in ISO 19772.

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

The TOE will provide cryptographic signature services using RSA Digital Signature Algorithm with key size of 2048 bits that meets the FIPS 186-4 Digital Signature Standard. The TOE's SSH implementation supports the following public key algorithms for public key-based authentication: ecdsa-sha2-nistp256; ecdsa-sha2-nistp384; and ecdsa-sha2-nistp521 meeting the FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and ISO/IEC 14888-3, Section 6.4 standard.

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

The TOE performs SHA-1, SHA-256, SHA-384, SHA-512 cryptographic hashing services in accordance with ISO/IEC 10118-3:2004. The SHA hash algorithm is used as part of HMAC, but is also used as part of RSA digital signature creation and verification. SHA-256, SHA-384, and SHA-512 also support ECDSA signature generation and verification.

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

The TOE performs keyed-hash message authentication that meets the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2. The key length, hash function used, block size, and output MAC lengths are identified in the table below.

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

Table 9 HMAC Properties

Keyed-hash message authentication services HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-512 are supported for SSH. SSH also uses implicit message authentication when the encryption algorithm is AES-GCM.

6.2.8 FCS_RBG_EXT.1: Random Bit Generation

The TOE uses a software-based deterministic random bit generator that complies with ISO/IEC 18031:2011, using CTR_DRBG (AES). The TOE seeds the DRBG with 256 bits of entropy. All platforms use entropy provided by the Linux kernel, including device, input, interrupt, and disk randomness.

On the 1100TX/5500TX/8400TX/8200TX and vTPS with a processor that supports the RDRAND instruction, RDRAND is used as an additional hardware based entropy input.

On the vTPS with a processor that does not support the RDRAND instruction, CPU time jitters are used as an additional non-physical based entropy input.

6.2.9 FCS_SHC_EXT.1 – SSH Client Protocol / FCS_SHS_EXT.1 – SSH Server Protocol

The TOE acts as an SSH client for secure communications with an external audit server. The TOE acts as an SSH Server for secure communications with remote administrators. The TOE implements the SSHv2 protocol and complies with RFCs 4251, 4252, 4253, 4254, 5647, 5656, and 6668. The TOE's server implementation also implements RFC 8268.

Both of the TOE's client and server implementations of SSH support the public-key-based authentication method as described in RFC 4252. The TOE's SSH server implementation also supports password-based authentication as described in RFC 4252. The TOE drops packets greater than 256K bytes in an SSH transport connection as described in RFC 4253.

As SSH packets are being received, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. However, if it is not complete when the buffer becomes full (256K bytes) the packet will be dropped.

The TOE's SSH transport implementation uses:

- aes128-cbc, aes256-cbc, aes128-gcm@openssh.com, aes256-gcm@openssh.com encryption algorithms
- ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384,and ecdsa-sha2-nistp521 as its public key algorithms; and
- hmac-sha1, hmac-sha2-256, hmac-sha2-512 (implicit for aes*-gcm@openssh.com) as its MAC algorithms.

The SSH algorithms are enabled by default. The TOE rejects any encryption, public key and MAC algorithms not listed above. SSH ciphers can be toggled using the command: **debug ssh ciphers CIPHER enable/disable**. PK and MAC algorithms can be changed by modifying the sshd config file as root.

The TOE's SSH client uses diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 as its key exchange methods. The TOE's SSH server uses diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, and diffie-hellman-group16-sha512 as its key exchange methods. The TOE ensures that the SSH connection is rekeyed when a threshold of either one hour has been reached, or when one gigabyte of data has been transmitted. Both thresholds are checked by the TOE and rekeying is performed upon reaching the threshold that is hit first.

The TOE requires users to be identified and authenticated before they can access any of the TOE functions.

6.3 Identification and Authentication

The TOE provides identification and authentication and password management functions.

6.3.1 FIA_AFL.1 Authentication Failure Handling

The TOE can detect when an Administrator (Super User and Admin) configurable number (from 1 to 10) of failed remote authentication attempts has been reached. When the defined number of unsuccessful authentication attempts has been reached, the remote administrator accessing the TOE via SSH is locked out for an administrator (Super User and Admin) configurable period of time (1-1440 minutes). Authentication failures by remote Administrators cannot lead to a situation where no Administrator access is available to the TOE. If remote administrators are locked out, administrator access is still available via local console. This prevents any condition where no administrator access is available.

6.3.2 FIA_PMG_EXT.1: Password Management

Passwords can be composed of any combination of upper and lower case letters, numbers, and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “,”, “.”, “/”, “<”, “>”, “?”. Single and double quotes, spaces or back slashes are not allowed. The minimum password length is administrator configurable to 1, 8 or 15 characters, depending on Password Security Level.

The TOE offers configurable global authentication settings that apply to all users. The TOE offers pre-defined Password Security Levels of None, Low, Medium and High. The default value is Medium. Each level adopts the requirements of the preceding level and adds additional requirements for the user name and password.

A Password Security Level of None does not contain any restrictions other than User names cannot contain spaces. The Password Security Level of Low requires User names to be at least six characters in length; a new password must be different than the current password, and passwords must be at least eight characters in length. A Password Security Level of Medium specifies the following additional password complexity requirements:

- Contains at least two alphabetic characters,
- Contains at least one numeric character, and
- Contains at least one non-alphanumeric character.

A Password Security Level of High requires the passwords to be at least 15 characters and meet the following additional password complexity requirements:

- Contains at least one uppercase character,
- Contains at least one lowercase character, and
- At least half the characters cannot occupy the same positions as the current password.

Based on the configured password security level, the only security-relevant condition is the enforced length configured by level.

6.3.3 FIA_UAU.7: Protected Authentication Feedback

When logging in, the TOE does not echo passwords so that passwords are not inadvertently displayed to the user and any other users that might be able to view the login display.

6.3.4 FIA_UIA_EXT.1: User Identification and Authentication, FIA_UAU_EXT.2: Password-based Authentication Mechanism

Administrators manage the TOE remotely using an SSH connection to the Ethernet Management port on the TOE appliance or locally through the console interface or direct connection to the Ethernet Management port. Each method provides access to the CLI after an administrator successfully logs in. Prior to administrative login, the Management interface will respond to ICMP requests to confirm connectivity (for remote administrative connections) and displays a warning banner for both local and remote connections. No other TSF-mediated actions are permitted on behalf of an administrative user until the user is successfully authenticated.

In order to log in, the user must provide an identity and also authentication data that matches an identity configured on the TOE. Users are defined locally within the TOE with a user identity, password, and user role. Administrators accessing the Ethernet Management port can be defined with an SSH public key for public key-based authentication for SSH connections rather than a password. Users are authenticated directly by the TOE. Any resulting session is dependent upon successful authentication and established sessions are associated with the role(s) (see Section 6.4) assigned to the user.

6.4 Security Management

The TOE provides a CLI to access the security management functions. Security management commands are limited to administrators and are available only after they have provided acceptable user identification and authentication data to the TOE. The TOE controls user access to the TOE and resources based on user role. Users are given permission to access a set of commands and resources based on their user role.

6.4.1 FMT_MOF.1/ManualUpdate: Management of Security Functions Behaviour Requests

The initiation of manual TOE updates is restricted to the Admin and Super User roles.

6.4.2 FMT_MOF.1/Functions: Management of Security Functions Behaviour

Users with the Super User, or Admin roles can configure the audit data to be transmitted to a remote syslog server.

6.4.3 FMT_MTD.1: Management of TSF Data

The ability to manage the TSF data is restricted to the Super User and Admin roles. The word 'manage' includes the following actions: create, initialize, view, change default, modify, and delete or clear. No administrative functions are accessible prior to administrator log-in. The authorized administrator must have the appropriate permissions as defined by the role to access the TSF data.

6.4.4 FMT_SMF.1/Core: Specification of Management Functions (Core)

All administrative functionality is available from the CLI (locally or remote).

The TOE provides the following management functions:

- Configure the access banner;
- Configure the cryptographic functionality (cryptographic ciphers used in SSH sessions);
- Set the time which is used for time-stamps;
- Update the TOE, and to verify the updates using the digital signature capability prior to installing those updates;
- Configure the authentication failure parameters for FIA_AFL.1;
- Configure the session inactivity time before session termination;
- Configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1; and
- Configure audit behaviour (set local log sizes; clear local log files).

6.4.5 FMT_SMF.1/IPS: Specification of Management Functions (IPS)

Authorized administrators use the CLI to manage the IPS functions. These management functions include:

- 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)
- Update (import) 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.

The IPS data analysis and reactions are configured using traffic management policies and filters as described in Section 6.8.

6.4.6 FMT_SMR.2: Restrictions on Security Roles

The TOE includes pre-defined administrator roles and supports local and remote administration. The pre-defined roles Super User, Admin, and Operator each provide a subset of administrative functions and collectively represent the Security Administrator role as described in the Security Problem Definition and in the Security Functional Requirements. The Operator role only has the ability to view TSF data as specified by FMT_MTD.1. Users assigned NORMAL_ACCESS permissions have no administrative permissions. The Super User, and Admin have full access to all IPS management functions and to all functions to manage the TOE as specified in FMT_SMF.1/IPS and FMT_SMF.1/Core.

The TPS appliance has a serial console interface as well as an Ethernet interface dedicated to management. An administrator can manage the TOE locally via the CLI through the console interface. In addition, the CLI can be accessed remotely via SSH.

6.5 Protection of the TSF

The TOE ensures that sensitive information such as passwords and cryptographic keys are stored such that they are not accessible even to an administrator. The TOE provides its own internal clock which it uses to provide a reliable time source for audit records.

The TOE includes functions to perform self-tests and mechanisms for the update of the TOE software/firmware and verification of the cryptographic functions.

6.5.1 FPT_APW_EXT.1: Protection of Administrator Passwords

The TOE stores administrative passwords using 256-bit AES and prevents reading of plaintext passwords. The TOE does not offer any functions that will disclose to any users a plaintext password. See Section 6.2 for more information about stored passwords.

6.5.2 FPT_SKP_EXT.1: Protection of TSF Data (for Reading of all Pre-shared, Symmetric and Private Keys)

The TOE does not offer any functions that will disclose to any users a stored cryptographic key. See Section 6.2 for more information about stored keys.

6.5.3 FPT_STM_EXT.1: Reliable Time Stamps

The TOE is a hardware appliance or a virtual appliance image installed on a hardware appliance that includes a hardware-based real-time clock to ensure that reliable time information is available. The TOE's real-time clock is a Complementary Metal-Oxide Semiconductor that stores the system time and date information. The TOE's embedded OS manages the clock and exposes administrator clock-related functions. The clock is used for audit record time stamps, measuring session activity for termination, and for cryptographic operations based on time/date. Additionally, the TOE synchronize time with an external NTP server.

6.5.4 FPT_TST_EXT.1: TSF Testing

The TOE performs all self-tests (software module integrity tests and cryptographic known answer tests) on start-up. The TOE process manager service is responsible for bringing up all relevant TOE processes. All binaries include an embedded integrity checksum (md5sum) that the process manager verifies before starting the process. If a module fails a software integrity test, the TOE reports status indicating which failure occurred and transitions to an error state, in which the module ceases to continue processing.

The TOE includes the OpenSSL 2.0.13 FIPS wrapper which forces the execution of the self-tests and ensures the correct operation of cryptographic functions. OpenSSL performs the following cryptographic self-tests during start-up:

- Cryptographic known answer tests: for symmetric and one-way cryptographic operations, the TSF will process known input data and compare it to the pre-computed output for each algorithm to ensure results are consistent with known answers.

- Pairwise consistency tests: for public key cryptographic operations, the TSF will perform a cryptographic operation followed by its reverse (e.g. encrypt/decrypt; sign/verify) to ensure that the result of the calculation is the same as the initially-supplied value.

6.5.5 FPT_TUD_EXT.1: Trusted Update

The administrator uses the CLI to update the TOE, and to query the currently executing software version of the TOE. The command **version** displays the current software version.

The administrator uses a Debug command (**debug upgrade**) to download a TOE update package directly from a specified URL. The update package is published on Trend Micro support website. The vendor generates a digital signature of the update package by first calculating the SHA-256 hash of the update package, then encrypting the generated hash using its 2048-bit RSA private key. The TOE update package includes the digital signature and the public key is included in the software image. The digital signature is verified by the TOE prior to the package being installed. The process is as follows: the TOE calculates its own SHA-256 hash of the update package, then decrypts the digital signature accompanying the update package using the RSA public key matching the vendor's private key, and comparing the hash it calculated with the decrypted hash value. If they are equal, the package is valid and has not been modified. The digital signature is downloaded as part of the update package, and the TOE is pre-installed with the public key. The TOE starts the update process once it verifies the signature/hash. A package with an invalid signature will not be installed by the TOE.

6.6 TOE Access

The TOE can be configured to display an informative banner when an administrator establishes an interactive session. The TOE can also enforce an administrator-defined inactivity timeout value after which the inactive session (local or remote) will be terminated. Finally, the TOE allows administrators to terminate their own session.

6.6.1 FTA_SSL.3: TSF-initiated Termination

The TOE can be configured by an administrator to set an interactive session timeout value (integer value from 1 to 32000) for local console and/or remote user sessions. The default timeout is 15 minutes.

Such sessions will be terminated when the inactivity period expires. Should a user have their session terminated (e.g., due to inactivity), they are required to successfully authenticate, by reentering their identity and authentication data, in order to establish a new session.

6.6.2 FTA_SSL.4: User-initiated Termination

Administrators can terminate their own interactive sessions by logging out at the console and SSH.

6.6.3 FTA_TAB.1: Default TOE Access Banners

The TOE supports an administrator-configurable TOE access banner that is displayed prior to a user completing the login process at the CLI. The banner text is configured using the **`login-banner`** command and the same text is displayed at both local and remote management connections (console, SSH).

6.7 Trusted Path/Channels

An authorized administrator can establish a secure remote connection with the TOE using SSH.

The TOE also uses SSH secure communications with an external log server to prevent unintended disclosure or modification of audit records.

6.7.1 FTP_ITC.1: Inter-TSF Trusted Channel

The TOE can be configured to export audit records to an external audit server. The TOE uses SSH to protect communications between itself and the audit server. SSH provides assured identification of its end points via host name/public key association as per FCS_SHC_EXT.1.9 and protection of the channel data from disclosure and detection of modification of the channel data. The TOE initiates communication via the trusted channel for the audit server.

The TOEs secure protocols are supported by FIPS Approved cryptographic mechanisms included in the TOE implementation.

6.7.2 FTP_TRP.1: Trusted Path

The TOE protects interactive communication with administrators accessing the CLI remotely using SSH, which provides confidentiality of transmitted information and detects any loss of integrity. Remote administrators initiate communication via the trusted path by using an SSH client to login.

To successfully establish an interactive administrative session, the administrator must be able to provide acceptable user credentials (e.g., user id and password), after which they will be able to access the CLI features. Remote administrators may alternatively need to provide an SSH key for key-based authentication. The trusted path is used for initial Administrator authentication and all subsequent administrative actions.

The secure protocols are supported by FIPS-approved cryptographic mechanisms included in the TOE implementation.

6.8 Intrusion Prevention System

The TOE's Threat Suppression Engine is a line-speed hardware engine that implements the Intrusion Prevention functions focusing on inspecting the IP traffic (TCP, UDP, ICMP, etc.). The TSE protects the network by scanning, detecting, and responding to network traffic according to the filters, action sets, and global settings maintained on each device.

The TOE uses Digital Vaccine (DV) filters to police the network and to screen out malicious or unwanted traffic. In addition to the DV filters, the TOE also provides Traffic Management filters, which are custom filters that react to traffic based on source IP address, destination IP address, port, protocol, or other defined values. Traffic management filters are applied to traffic before DV filters. Each DV package has a unique ID that the DV distribution service uses to distinguish different DV packages. Each DV filter has a Category, an Action set, and State and is customizable to be enabled or disabled and to identify an action set. The **Category** component defines the type of network protection provided by the filter (e.g. exploits, security-policy, spyware, virus). The category is also used to control the global filter settings using the Category Setting configuration. The filter categories can be enabled or disabled and action sets can be applied to the filters using the **`category-settings`** command. If a category is disabled, all filters in the category are disabled. When a filter is disabled, it is not applied to traffic. Limits and exceptions change the way filters are applied based on IP address. For example, a limit setting can be specified so that filters only apply to specific source and destination IP addresses or address ranges. If a filter has both global and filter level exception settings, the Threat Suppression Engine uses the filter-level settings to determine how to apply the filter. The following limit and exceptions can be configured:

- **Filter Exceptions** (specific) — Allow traffic that would normally trigger a filter to pass between specific addresses or address ranges without triggering the filter. Configured using the **`Filter`** command, these exceptions apply only to the filter where they were configured.
- **Limit Filter to IP Addresses** (global) — Only apply filters to traffic between specified source and destination IP address pairs. IP address limits can be configured that apply to Application Protection, Traffic Normalization, and Network Equipment Protection filter types. Separate limits can be configured that apply only to Performance Protection filters.
- **Exceptions** (global) — Exclude traffic between specified source and destination IP address pairs. Exceptions can be configured for the following filter types: Application Protection, Traffic Normalization, Network Equipment Protection, and Performance Protection filters. These exceptions are global for all specified filters.

Each filter also has an Adaptive Filter Configuration State component that allows the global Adaptive Filter configuration to be set to over-ride so that the filter is not affected by adaptive filtering. Adaptive filtering works by monitoring each filter to identify any suspected of causing congestion. When it identifies a filter, it manages the filter using one of the following methods, depending on how the global or filter-level Adaptive Filtering is configured:

- **Automatic Mode** — This setting enables the IPS device to automatically disable and generate a system message regarding the defective filter.
- **Manual** — This setting enables the IPS device to generate a system message regarding the defective filter. However, the filter is not disabled.

The following IPS management configurations are also available from the console:

- Modify the duration of traffic blocking actions: Edit >>ips >>connection-table
- Configure the known-good and known-bad lists to override signature-based IPS policies
  - Edit>>ips>>profile "profile name" >>exception (for policy level exceptions)
  - Edit>>ips>>profile "profile name" >>filter " filter number" >>exception (for signature based exceptions)

A security profile defines the traffic that the IPS monitors and the signature-based DV filters that the IPS applies. Traffic monitoring is based on incoming and outgoing port pairs. The administrator can use the default DV filter configuration to protect the segment or customize the configuration as required. The segment specifies both the port and the traffic direction, which allows definition of separate security profiles for traffic in and out of a port. Traffic management filters are configured in the context of a traffic management profile that determines which network segments are monitored by the filter. The Traffic management profile defines IP layer parameter configuration to control packet flows. If a security profile is not specified for the filter, the filter is applied to the Default security profile.

The default security profile is set to ANY incoming ports and ANY outgoing ports, with all IPS filters configured with the default Digital Vaccine settings. With the default profile in place, all incoming and outgoing traffic on any virtual segment configured on the device is monitored according to the IPS filter configuration recommended by TippingPoint. The administrator can customize the default security profile with the virtual segments that it applies to, modify the filter settings, or modify the security profiles as required. The **`profile`** command can be sued to create, modify, or delete security or traffic management profiles. The subcommand **`traffic-mgmt`** can be used to create a traffic management profile.

6.8.1 IPS_ABD_EXT.1: Anomaly-Based IPS Functionality

The TOE supports anomaly traffic patterns for inline traffic by providing the ability to define rate-limit policy filters that specify throughput thresholds in Kbps that can be applied to the traffic. The thresholds define the frequency at which traffic (in kilobytes) can traverse the TOE (per second) before the filter is triggered. Traffic that deviates from the defined throughput thresholds is considered unexpected or atypical and treated as potentially malicious. Filters for anomaly-based policies can be applied to any of the network protocol fields (all packet header and data elements defined in IPS_SBD_EXT.1. Each rule can be associated with any of the following operations for inline traffic using the **`action-set`** command:

- allow the traffic flow
- block/drop 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, destination, port) unreachable message.

Rate-limit policy filters are defined in traffic management profiles using the **`rate-limit`** subcommand to create/modify an action (as defined above) that rate-limits traffic according to the identified threshold (defined in Kbps). Traffic matching a rate-limit filter is subsequently inspected based on the security profile configuration (DV filtering). That is, traffic is not allowed through the device based solely on the rate-limit traffic management filter criteria. Rate-limit policy filters rules can be applied and enforced on any of the TOE’s sensor interfaces. Different TPS appliances support different sensor interfaces ranging from 1GbE to 40GbE.

DV filters are included in the system as the place-holder for users to define the frequency and threshold values identified above as used in anomaly-based detection and prevention.

Traffic matching the manually configured anomaly-based rules (rate-limit) are subsequently inspected based on the security profile configuration (DV filtering).

6.8.2 IPS_IPB_EXT.1: IP Blocking

The TOE supports configuration and implementation of known-good and known-bad lists of source and destination IP addresses. Configuration of the policy elements is restricted to the IPS Administrators (i.e. users with the Super User or Admin role). Known-bad lists are configured and provided through traffic management filters. Known-bad lists and additionally known-good lists can be configured in Reputation filters within a Security Policy using the **`reputation`** and **`reputation groups`** commands that allows an administrator to create groups of IPv4, IPv6, and DNS addresses, and apply block, permit, or notify actions across an entire reputation group. After a group is configured, security profiles can be configured to apply reputation filters to the group. Additional inspection is performed after Reputation checks by the DV Filters. Overrides to the additional signature based policy inspection can be implemented from the console >>Edit>>IPS>>profile "profilename" >>filter "filter number" >>Exception.

When an IP address or DNS name is added to a reputation group, it is added to the device's reputation database. Incoming traffic is checked against the database, and the appropriate reputation filters are then applied. Traffic management filters are configured in the context of a traffic management profile that determines which network segments are monitored by the filter. If a security profile is not specified for the filter, the filter is applied to the Default security profile.

The TOE provides the Reputation ThreatDV package that is separate from the standard DV filter package. This package includes pre-defined known-bad lists. Reputation ThreatDV package updates can be obtained in the same manner as the standard DV packages as described in Section 6.8.4.

6.8.3 IPS_NTA_EXT.1: Network Traffic Analysis

The TOE performs analysis of IP-based network traffic forwarded to the TOE's sensor interfaces, and detects violations of administratively-defined IPS policies. The TOE is able inspect and process the following network protocols: IPv4 (RFC 791); IPv6 (RFC 2460); ICMPv4 (RFC 792); ICMPv6 (RFC 2463); TCP (RFC 793); and UDP (RFC 768). Trend Micro determines protocol conformance through developer testing in the context of filter behavior.

The TOE provides approximately 20,000 filters that are the basis for its signature-based IPS functionality. Not all filters are turned on simultaneously. Some filters apply to perimeter TPS devices while others are for interior TPS devices. The TOE provides pre-packaged profiles (for example standard, aggressive, hyper-aggressive). Each pre-packaged profile balances filtering and false policy. New filters are published on a regular basis and customers are provided with descriptions and guidance for the new filters. Filters are preset to on or off in filter profiles, but an administrator can enable/disable as needed. The filter profiles (containing signatures) can be assigned to the TOE's sensor interfaces configured for inline mode (1, 10, 40 GB Ethernet data interfaces, virtio or vmxnet3 for vTPS) and supports designation of both a 1 Gb Ethernet (SSH access to CLI) and a serial interface (CLI) as management modes for communication between the TOE and external entities without simultaneously being sensor interfaces. Promiscuous mode is not supported.

The TOE's policy hierarchy (precedence) is as follows: reputation filters are applied between anomaly-based filters and signature-based filters. The order of these filter matching operations cannot be changed except for signature-based filters, which have a pre-defined precedence; all other types of filters are applied in the order of their definition.

6.8.4 IPS_SBD_EXT.1: Signature-Based IPS Functionality

Trend Micro provides approximately 10,000 filters. Not all filters are turned on simultaneously. Some filters apply to perimeter TOE devices while others are for interior TPS devices. Trend Micro provides pre-packaged profiles (for example: standard, aggressive, hyper-aggressive). Trend Micro publishes new filters to their Threat Management Center (TMC) on a regular basis and sends descriptions and guidance for the new filters to customers. Filters are preset to on or off in filter profiles, but an administrator can enable/disable as needed. Filters can be applied by interface. The command **`conf t autodv`** enables and disables the automatic download service for Digital Vaccine updates. This command requires a day of week and time of day for the download. If desired, the administrator can use the **`-period`** option to set the number of days between checks.

A string-based detection signature or rule is comprised of the string or the attack's signature to be matched with the packet's payload. The TOE reconstructs and inspects flow payloads by parsing the traffic at the application layer. As each new packet of the traffic flow arrives, the engine re-evaluates the traffic for malicious content. The instant the engine detects malicious traffic, it blocks all current and all subsequent packets pertaining to the traffic flow. The blocking of the traffic and packets ensures that the attack never reaches its destination.

The TOE is capable of inspecting the following packet header fields:

- IPv4: version; header length; packet length; ID; IP flags; fragment offset; time to live (ttl); protocol; header checksum; source address; destination address; and IP options.
- IPv6: version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options.
- ICMP: type; code; header checksum; ID; and sequence number.
- ICMPv6: type; code; and header checksum
- TCP: source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.
- UDP: source port; destination port; length; and UDP checksum.

The TOE is capable of inspecting 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:
  - 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.
  - HTTP (web) commands and content: commands including GET and POST, and administrator-defined strings to match URLs/URIs, and web page content.
  - SMTP (email) states: start state, SMTP commands state, mail header state, mail body state, abort state.
- UDP data: characters beyond the first 8 bytes of the UDP header

Additionally, the TOE supports stream reassembly for detecting malicious payload even if it is split across multiple non-fragmented packets.

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

- IP Attacks
  - IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack)
  - IP source address equal to the IP destination (Land attack)
- ICMP Attacks
  - Fragmented ICMP Traffic (e.g. Nuke attack)
  - Large ICMP Traffic (Ping of Death attack)
- TCP Attacks
  - TCP NULL flags
  - TCP SYN+FIN flags
  - TCP FIN only flags
  - TCP SYN+RST flags
- UDP Attacks
  - UDP Bomb Attack
  - UDP Chargen DoS Attack

The TOE processes and triggers reactions for the attack signatures as follows. For every network flow defined by 5-tuple (source and destination IP address and port plus transport protocol, e.g. ICMP, TCP and UDP), the TOE keeps the protocol state and statistics of the flow. Filters can be used to define the conditions under which the flow will be triggered and defined policy will be applied. For example, a “IP Fragments Overlap” filter can be used to trigger flows that include overlapped IP fragments and policy can be define as “block” to prevent such flows.

The TOE is able to detect and apply the following traffic-pattern detection signatures to IPS sensor interfaces:

- Flooding a host (DoS attack)
- ICMP flooding (Smurf attack, and ping flood)
- TCP flooding (e.g. SYN flood)
- Flooding a network (DoS attack)
- IP Protocol and TCP, UDP, ICMP port scanning

For port scanning, during the inspection process, the TOE first scans traffic against the standard ports for listed services, and then scans traffic against the list of additional ports. When an anomaly-based filter is defined and enabled, the TOE will track and keep statistics associated with the defined parameters: IP addresses, ports and protocols, e.g. ICMP, UDP and TCP. Frequency and threshold will then be calculated and compared to trigger the policy application.

The TOE supports inline mode only. In this mode, the TOE is able to allow or block/drop traffic flows and can send TCP resets to source and destination TCP addresses. Any of these actions can be applied to each signature-based filter.

7. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives
- Security Functional Requirements
- Security Assurance Requirements
- Requirement Dependencies
- TOE Summary Specification.

7.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, policy or threat.

7.1.1 Security Objectives Rationale for the TOE

This section shows that all threats and OSPs are completely covered by security objectives for the TOE. In addition, each objective counters or addresses at least one threat or OSP.

Table 10 Threats and OSPs to TOE Security Objectives Correspondence

	P.ANALYZE	T.NETWORK_ACCESS	T.PASSWORD_CRACKING	T.NETWORK_DISCLOSURE	T.NETWORK_DOS	T.NETWORK_MISUSE	T.SECURITY_FUNCTIONALITY_COMPROMISE	T.SECURITY_FUNCTIONALITY_FAILURE	T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	T.UNDETECTED_ACTIVITY	T.UNTRUSTED_COMMUNICATION_CHANNELS	T.UPDATE_COMPROMISE	T.WEAK_AUTHENTICATION_ENDPOINTS	T. WEAK_CRYPTOGRAPHY	P.ACCESS_BANNER
O.AUDIT_GENERATION										X					
O.DISPLAY_BANNER									X						X
O.IPS_ANALYZE	X	X		X	X	X									
O.IPS_REACT		X		X	X	X									
O.PROTECTED_COMMUNIATIONS											X		X		
O.SECURITY_FUNCTIONALITY_COMPROMISE							X								
O.STRONG_CRYPTOGRAPHY											X	X	X	X	

	P.ANALYZE	T.NETWORK_ACCESS	T.PASSWORD_CRACKING	T.NETWORK_DISCLOSURE	T.NETWORK_DOS	T.NETWORK_MISUSE	T.SECURITY_FUNCTIONALITY_COMPROMISE	T.SECURITY_FUNCTIONALITY_FAILURE	T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	T.UNDETECTED_ACTIVITY	T.UNTRUSTED_COMMUNICATION_CHANNELS	T.UPDATE_COMPROMISE	T.WEAK_AUTHENTICATION_ENDPOINTS	T. WEAK_CRYPTOGRAPHY	P.ACCESS_BANNER
O.SYSTEM_MONITORING		X		X	X	X									
O.TOE_ADMINISTRATION	X		X						X						
O.TSF_SELF_TEST								X							
O.VERIFIABLE_UPDATES												X			

T.NETWORK_ACCESS

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..

This threat is countered by ensuring that:

- O.IPS_ANALYZE: Entities that reside on or communicate across monitored networks must have network activity effectively analyzed for potential violations of approved network usage. The TOE is able to effectively analyze data collected from monitored networks to reduce the risk of unauthorized disclosure of information, inappropriate access to services, and misuse of network resources.
- O.IPS_REACT: The TOE is able to react in real-time as configured by the Security Administrator to terminate and block traffic flows that have been determined to violate administrator-defined IPS policies.
- O.SYSTEM_MONITORING: The TOE is able to analyze and react to potential network policy violations and collect and store essential data elements of network traffic on monitored networks.

T.PASSWORD_CRACKING

Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device.

This threat is countered by ensuring that:

- O.TOE_ADMINISTRATION: The TOE will provide mechanisms to ensure that only administrators are able to log in and use the management interfaces to configure the TOE and its network, mechanisms that control a user's logical access to the TOE and mechanisms to ensure strong passwords.

T.NETWORK_DISCLOSURE

Sensitive information on a protected network might be disclosed to an attacker resulting from ingress- or egress-based actions.

This threat is countered by ensuring that:

- O.IPS_ANALYZE: Entities that reside on or communicate across monitored networks must have network activity effectively analyzed for potential violations of approved network usage. The TOE is able to effectively analyze data collected from monitored networks to reduce the risk of unauthorized disclosure of information, inappropriate access to services, and misuse of network resources.
- O.IPS_REACT: The TOE is able to react in real-time as configured by the Security Administrator to terminate and block traffic flows that have been determined to violate administrator-defined IPS policies.
- O.SYSTEM_MONITORING: The TOE is able to analyze and react to potential network policy violations, as well as collect and store essential data elements of network traffic on monitored networks.

T.NETWORK_DOS

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.

This threat is countered by ensuring that:

- O.IPS_ANALYZE: Entities that reside on or communicate across monitored networks must have network activity effectively analyzed for potential violations of approved network usage. The TOE is able to effectively analyze data collected from monitored networks to reduce the risk of unauthorized disclosure of information, inappropriate access to services, and misuse of network resources.
- O.IPS_REACT: The TOE is able to react in real-time as configured by the Security Administrator to terminate and block traffic flows that have been determined to violate administrator-defined IPS policies.
- O.SYSTEM_MONITORING: The TOE is able to analyze and react to potential network policy violations, as well as collect and store essential data elements of network traffic on monitored networks.

T.NETWORK_MISUSE

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, forced resets, malicious zip files, disguised executables, privilege escalation tools, and botnets)..

This threat is countered by ensuring that:

- O.IPS_ANALYZE: Entities that reside on or communicate across monitored networks must have network activity effectively analyzed for potential violations of approved network usage. The TOE is able to effectively analyze data collected from monitored networks to reduce the risk of unauthorized disclosure of information, inappropriate access to services, and misuse of network resources.
- O.IPS_REACT: The TOE is able to react in real-time as configured by the Security Administrator to terminate and block traffic flows that have been determined to violate administrator-defined IPS policies.
- O.SYSTEM_MONITORING: The TOE is able to analyze and react to potential network policy violations, as well as collect and store essential data elements of network traffic on monitored networks.

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.

This threat is countered by ensuring that:

- O.SECURITY_FUNCTIONALITY_COMPROMISE: The TOE destroys cryptographic keys when they are no longer needed and prevents the reading of pre-shared keys, symmetric keys, and private keys.

T.SECURITY_FUNCTIONALITY_FAILURE

A component of the TOE may fail during start-up or during operations causing a compromise or failure in the security functionality of the TOE, leaving the TOE unavailable and/or susceptible to attackers.

This threat is satisfied by ensuring that:

- O.TSF_SELF_TEST: The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

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.

This threat is countered by ensuring that:

- O.DISPLAY_BANNER: The TOE will display an advisory warning regarding use of the TOE.
- O.TOE_ADMINISTRATION: The TOE will provide mechanisms to ensure that only administrators are able to log in and use the management interfaces to configure the TOE and its network, mechanisms that control a user's logical access to the TOE and mechanisms to ensure strong passwords.

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.

This threat is countered by ensuring that:

- O.AUDIT_GENERATION: The TOE will provide the capability to detect and create records of security relevant events associated with users; and store those audit data locally, or externally if configured.

T.UNTRUSTED_COMMUNICATION_CHANNELS

Threat agents may attempt to target network devices that do not use standardized secure tunneling 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.

This threat is countered by ensuring that:

- O. PROTECTED_COMMUNICATIONS: The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
- O.STRONG_CRYPTOGRAPHY: The TOE will provide strong standards-based cryptographic algorithms and key sizes.

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.

This threat is countered by ensuring that:

- O.VERIFIABLE_UPDATES: The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and cryptographically validated to be from a trusted source.
- O.STRONG_CRYPTOGRAPHY: The TOE will provide strong standards-based cryptographic algorithms and key sizes.

T. WEAK_AUTHENTICATION_ENDPOINTS

Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., shared password that is guessable or transported as plaintext.

This threat is countered by ensuring that:

- O. PROTECTED_COMMUNICATIONS: The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
- O.STRONG_CRYPTOGRAPHY: The TOE will provide strong standards-based cryptographic algorithms and key sizes.

T. WEAK_CRYPTOGRAPHY

Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space.

This threat is countered by ensuring that:

- O.STRONG_CRYPTOGRAPHY: The TOE will provide strong standards-based cryptographic algorithms and key sizes.

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.

This policy is covered by ensuring that:

- O.DISPLAY_BANNER: The TOE will display an advisory warning regarding use of the TOE.

P.ANALYZE

Analytical processes and information to derive conclusions about potential intrusions must be applied to IPS data and appropriate response actions taken.

This policy is covered by ensuring that:

- O.IPS_ANALYZE: Entities that reside on or communicate across monitored networks must have network activity effectively analyzed for potential violations of approved network usage. The TOE will effectively analyze data collected from monitored networks to reduce the risk of unauthorized disclosure of information, inappropriate access to services, and misuse of network resources.

7.1.2 Security Objectives Rationale for the Operational Environment

This section shows that all secure usage assumptions are completely covered by security objectives for the operational environment of the TOE. In addition, each objective addresses at least one assumption.

Table 11 Assumptions and Policies to Operational Environment Security Objectives Correspondence

	A.ADMIN_CREDENTIALS_SECURE	A.CONNECTIONS	A.LIMITED_FUNCTIONALITY	A. TRUSTED_ADMINISTRATOR	A.PHYSICAL_PROTECTION	A. .REGULAR_UPDATES
OE.ADMIN_CREDENTIALS_SECURE	X					
OE.CONNECTIONS		X				
OE. NO_GENERAL_PURPOSE			X			
OE.PHYSICAL					X	
OE. TRUSTED_ADMIN				X		
OE.UPDATES						X

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.

This assumption is addressed by ensuring that:

- **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.

A.CONNECTIONS

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.

This assumption is addressed by ensuring that:

- **OE.CONNECTIONS:** 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.

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 computing platform for general purpose applications (unrelated to networking functionality).*

This assumption is addressed by ensuring that:

- 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.

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 and/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.*

This assumption is addressed by ensuring that:

- OE.PHYSICAL: Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

A. TRUSTED_ADMINISTRATOR

> *The Authorized 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 being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The 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.*

This assumption is addressed by ensuring that:

- OE. TRUSTED_ADMIN: TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

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.*

This assumption is addressed by ensuring that:

- 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.

7.2 Security Requirements Rationale

All security functional requirements identified in this Security Target are fully addressed in this section and each is mapped to the objective it is intended to satisfy. Table 15 summarizes the correspondence of functional requirements to TOE security objectives.

7.2.1 Security Functional Requirements Rationale

All of the Security Functional Requirements (SFRs) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective it is intended to satisfy.

Table 12 Objectives to Requirements Correspondence

	O.AUDIT_GENERATION	O.DISPLAY_BANNER	O.IPS_ANALYZE	O.IPS_REACT	O.PROTECTED_COMMUNICATIONS	O.SECURITY_FUNCTIONALITY_COMPROMISE	O.STRONG_CRYPTOGRAPHY	O.SYSTEM_MONITORING	O.TOE_ADMINISTRATION	O.TSF_SELF_TEST	O.VERIFIABLE_UPDATES
FAU_GEN.1/Audit	X										
FAU_GEN.1/IPS	X							X			
FAU_GEN.2	X										
FAU_STG.1/Audit	X										
FAU_STG.1/IPS	X							X			
FAU_STG_EXT.1	X										
FAU_STG_EXT.3	X										
FCS_CKM.1					X		X				
FCS_CKM.2					X		X				
FCS_CKM.4					X	X	X				
FCS_COP.1/ DataEncryption					X		X				
FCS_COP.1/SigGen					X		X				X
FCS_COP.1/Hash					X		X				
FCS_COP.1/KeyedHash					X		X				
FCS_RBG_EXT.1					X		X				
FCS_SHC_EXT.1					X		X				
FCS_SHS_EXT.1					X		X				
FIA_AFL.1									X		
FIA_PMG_EXT.1									X		
FIA_UIA_EXT.1									X		
FIA_UAU_EXT.2									X		

	O.AUDIT_GENERATION	O.DISPLAY_BANNER	O.IPS_ANALYZE	O.IPS_REACT	O.PROTECTED_COMMUNICATIONS	O.SECURITY_FUNCTIONALITY_COMPROMISE	O.STRONG_CRYPTOGRAPHY	O.SYSTEM_MONITORING	O.TOE_ADMINISTRATION	O.TSF_SELF_TEST	O.VERIFIABLE_UPDATES
FIA_UAU.7									X		
FMT_MOF.1/ManualUpdate									X		
FMT_MOF.1/Functions									X		
FMT_MTD.1									X		
FMT_SMF.1/Core									X		
FMT_SMF.1/IPS									X		
FMT_SMR.2									X		
FPT_APW_EXT.1									X		
FPT_SKP_EXT.1						X					
FPT_STM_EXT.1	X										
FPT_TST_EXT.1										X	
FPT_TUD_EXT.1											X
FTA_SSL.3									X		
FTA_SSL.4									X		
FTA_TAB.1		X									
FTP_ITC.1					X						
FTP_TRP.1					X						
IPS_ABD_EXT.1			X	X							
IPS_IPB_EXT.1			X								
IPS_NTA_EXT.1			X								
IPS_SBD_EXT.1			X	X							

O.AUDIT_GENERATION

- *The TOE will provide the capability to detect and create records of security relevant events associated with users; store those audit data locally, or externally if configured; and provide a means to control and protect how the audit records are handled when the local storage space for audit data is full.*

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1/Audit: The TOE is required to provide a set of events that it is capable of recording. Among these events the TOE is able to audit must be security relevant events occurring within the TOE. This requirement also defines the information that must be recorded for each auditable event.
- FAU_GEN.1/IPS: The TOE is required to generate audit events specific to IPS functionality. This requirement also defines the IPS related network traffic audit data that the TSF is capable of collecting.
- FAU_GEN.2: The TOE is required to associate a user identity resulting from actions of identified users with the identity of the user that caused the event.
- FAU_STG_EXT.1: The TOE is required to provide the ability to transmit audit data to a remote syslog server over SSH; store audit data locally; and control how the audit records are handled when the local storage space for audit data is full.
- FAU_STG_EXT.3: The TOE is required to generate a warning to inform the Administrator before the audit trail exceeds the local audit trail storage capacity.
- FAU_STG.1/Audit: The TOE is required to protect the stored audit records in the audit trail from unauthorized deletion and prevents unauthorized modifications to the stored audit records in the audit trail.
- FAU_STG.1/IPS: The TOE is required to protect the stored IPS audit records in the audit trail from unauthorized deletion and prevents unauthorized modifications to the stored IPS audit records in the audit trail.
- FPT_STM_EXT.1: The TOE is required to provide reliable time stamps for its own use. The timestamps are used in the audit function. The TOE is required to provide management functions that allow the Security Administrator to set the time and additionally can synchronize time with an external NTP server.

O. DISPLAY_BANNER

The TOE will display an administrator-configurable advisory warning regarding use of the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FTA_TAB.1: Before establishing a user session, the TOE is required to display an administrator-configurable advisory warning message regarding unauthorized use of the TOE.

O.IPS_ANALYZE

Entities that reside on or communicate across monitored networks must have network activity effectively analyzed for potential violations of approved network usage. The TOE must be able to effectively analyze data collected from monitored networks to reduce the risk of unauthorized disclosure of information, inappropriate access to services, and misuse of network resources.

This TOE Security Objective is satisfied by ensuring that:

- IPS_ABD_EXT.1: The TOE supports the definition of anomaly traffic patterns and is able to analyze network traffic for anomalous behavior that could indicate malicious activity.
- IPS_IPB_EXT.1: The TOE supports traffic analysis representing known-good and known-bad activities based on IP address.
- IPS_NTA_EXT.1: The TOE provides the ability to analyze network traffic based on supported protocols and network architecture characteristics.
- IPS_SBD_EXT.1: The TOE is able to detect potential malicious activity based on packet signatures.

O.IPS_REACT

The TOE must be able to react in real-time as configured by the Security Administrator to terminate and block traffic flows that have been determined to violate administrator-defined IPS policies.

This TOE Security Objective is satisfied by ensuring that:

- IPS_ABD_EXT.1: The TOE is able to specify the TOE's reaction to the detection of anomalous network traffic as configured by the Security Administrator.
- IPS_SBD_EXT.1: The TOE provides the ability to specify the TOE's reaction to the detection of an IPS signature in processed network traffic.

O.PROTECTED_COMMUNICATIONS

The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.1: The TOE is required to generate asymmetric cryptographic keys for use in key establishment. These keys meet the recommendations of FIPS PUB 186-4 or NIST Special Publication 800-56A Revision 3.
- FCS_CKM.2: The TOE is required to provide key distribution methods, specifically cryptographic key establishment in accordance with specified standards for secure trusted SSH channels.
- FCS_CKM.4: The TOE is required to clear (or destroy/zeroize), by overwriting with zeros, plaintext secret and private cryptographic keys and Cryptographic Critical Security Parameters (CSPs) when no longer required.
- FCS_COP.1/DataEncryption: The TOE is required to implement AES for encryption and decryption of data according to specific standards; CBC and GCM modes with 128/256-bit key sizes for OpenSSL used for SSH.
- FCS_COP.1/SigGen: The TOE is required to performs RSA Digital Signature Algorithm (rDSA) with cryptographic key sizes 2048 bits or greater that meet FIPS PUB 186-4. The TOE is required to perform Elliptic Curve Digital Signature Algorithm with cryptographic key sizes 256 bits, 384-bits, 521-bits that meet FIPS PUB 186-4.
- FCS_COP.1/Hash: The TOE is required to implement SHA-1, SHA-256, SHA-384, SHA-512 for hashing services as described above that meet ISO/IEC 10118-3:2004 with the required message digest sizes.
- FCS_COP.1/KeyedHash: The TOE is required to implement HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-512 for keyed-hash authentication that meet ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2" with the required key sizes.
- FCS_RBG_EXT.1: The TOE is required to implement CTR-DRBG(AES) with a minimum entropy seed of 256-bits that meet ISO/IEC 18031:2011.
- FCS_SHC_EXT.1: The TOE is required to implement the SSH Client Protocol.
- FCS_SHS_EXT.1: The TOE is required to implement the SSH Server Protocol.
- FTP_ITC.1: The TOE is required to utilize SSH to protect all data transmitted between the TOE and trusted external third-party IT entities from unauthorized disclosure and detection of modification during transmission.
- FTP_TRP.1: The TOE requires remote Administrators to connect to the TOE using SSH in order to use the administrative CLI for management of the TOE. The initial administrator authentication operation, as well as all subsequent remote administration actions, occurs through this channel.

O.SECURITY_FUNCTIONALITY_COMPROMISE

The TOE will properly destroy cryptographic keys when they are no longer needed and will prevent the reading of pre-shared keys, symmetric keys, and private keys.

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.4: The TOE is required to clear (or destroy/zeroize), by overwriting with zeros, plaintext secret and private cryptographic keys and Cryptographic Critical Security Parameters (CSPs) when no longer required.

- FPT_SKP_EXT.1: The TOE is required to prevent the reading of pre-shared keys, symmetric keys, and private keys.

O.STRONG_CRYPTOGRAPHY

The TOE will provide strong standards-based cryptographic algorithms and key sizes.

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.1: The TOE is required to generate asymmetric cryptographic keys for use in key establishment. These keys meet the recommendations of FIPS PUB 186-4 or NIST Special Publication 800-56A Revision 3.
- FCS_CKM.2: The TOE is required to provide key distribution methods, specifically cryptographic key establishment in accordance with specified standards for secure trusted SSH channels.
- FCS_CKM.4: The TOE is required to clear (or destroy/zeroize), by overwriting with zeros, plaintext secret and private cryptographic keys and Cryptographic Critical Security Parameters (CSPs) when no longer required.
- FCS_COP.1/DataEncryption: The TOE is required to implement AES for encryption and decryption of data according to specific standards; CBC and GCM modes with 128/256-bit key sizes for OpenSSL used for SSH.
- FCS_COP.1/SigGen: The TOE is required to performs RSA Digital Signature Algorithm (rDSA) with cryptographic key sizes 2048 bits or greater that meet FIPS PUB 186-4. The TOE is required to perform Elliptic Curve Digital Signature Algorithm with cryptographic key sizes 256 bits, 384-bits, 521-bits that meet FIPS PUB 186-4.
- FCS_COP.1/Hash: The TOE is required to implement SHA-1, SHA-256, SHA-384, SHA-512 for hashing services as described above that meet ISO/IEC 10118-3:2004 with the required message digest sizes.
- FCS_COP.1/KeyedHash: The TOE is required to implement HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-512 for keyed-hash authentication that meet ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2” with the required key sizes.
- FCS_RBG_EXT.1: The TOE is required to implement CTR-DRBG(AES) with a minimum entropy seed of 256-bits that meet ISO/IEC 18031:2011.
- FCS_SHC_EXT.1: The TOE is required to implement the SSH Client Protocol.
- FCS_SHS_EXT.1: The TOE is required to implement the SSH Server Protocol.

O. SYSTEM_MONITORING

To be able to analyze and react to potential network policy violations, the IPS must be able to collect and store essential data elements of network traffic on monitored networks.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1/IPS,: The TOE is required to generate audit events specific to IPS functionality. This requirement also defines the IPS related network traffic audit data that the TSF is capable of collecting.
- FAU_STG.1/IPS: This SFR supports this objective by requiring the TOE to protect the stored IPS audit records in the audit trail from unauthorized deletion and modifications to the stored IPS audit records in the audit trail.

O.TOE_ADMINISTRATION

The TOE will provide mechanisms to ensure that only administrators are able to log in and use the management interfaces to configure the TOE and its network, provide mechanisms that control a user’s logical access to the TOE, provide mechanisms to configure IPS capabilities; and mechanisms to ensure strong passwords.

This TOE Security Objective is satisfied by ensuring that:

- FIA_AFL.1: The TOE is required to provide an Authentication Failure Handling mechanism.

- FIA_PMG_EXT.1: The TOE is required to provide a mechanism to verify that secrets meet a defined quality metric.
- FIA_UAU_EXT.2: The TOE is required to provide a locally based administrative user authentication mechanism before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UIA_EXT.1: The TOE requires all administrative users to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that user allowing for only specific actions to be exempted from the authentication process.
- FIA_UAU.7: The TOE is required to provide only obscured feedback to the user while the authentication is in progress.
- FMT_MOF.1/ManualUpdate: The TOE is required to restrict the ability to enable the functions to perform manual updates to Security Administrators.
- FMT_MOF.1/Functions: The TOE is required to restrict the ability to determine and modify the behavior of the transmission of audit data to an external IT entity to Security Administrators.
- FMT_MTD.1: The TOE is required to ensure that only Security Administrators can manage the TSF data.
- FMT_SMF.1/Core: The TOE is required to provide management functions to support an administrator's ability to securely operate and configure the TOE and its environment.
- FMT_SMF.1/IPS: The TOE is required to provide management functions to support an administrator's ability to securely operate and configure the TOE's IPS functionality.
- FMT_SMR.2: The TOE is required to maintain the Security Administrator role; be capable of associating users with roles; and provide these users with local and remote administrative capabilities.
- FPT_APW_EXT.1: The TOE is required to protect administrator passwords by storing them encrypted and prevents reading of plaintext passwords by not providing any means to read them.
- FTA_SSL.3: The TOE is required to terminate local and remote interactive sessions after an administrator configurable time interval.
- FTA_SSL.4: The TOE is required to provide interfaces that allow administrator-initiated termination of the administrator's own interactive session.

O. TSF_SELF_TEST

The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

This TOE Security Objective is satisfied by ensuring that:

- FPT_TST_EXT.1: The TOE is required to implement self-tests, during initial startup, to determine whether the TOE is operating correctly.

O. VERIFIABLE_UPDATES

The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and cryptographically validated to be from a trusted source.

This TOE Security Objective is satisfied by ensuring that:

- FPT_TUD_EXT.1: The TOE is required to provide administrators the ability to: query the current version of the TOE firmware/software; and initiate updates to TOE firmware/software. The TOE provides a digital signature mechanism to verify firmware/software updates to the TOE prior to installing those updates.
- FCS_COP.1/SigGen: The TOE is required to perform RSA Digital Signature Algorithm (rDSA) with SHA-256 hash and 2048-bit RSA private key.

7.2.2 Security Assurance Requirements Rationale

The security assurance requirements for the TOE are the EAL 2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components. EAL 2 was selected as the assurance level because the TOE is a commercial product whose users require a low to moderate level of independently assured security.

The TOE is intended for use in an environment with good physical access security where it is assumed that attackers will have Basic attack potential. The target assurance level of EAL 2 is appropriate for such an environment.

7.3 Requirements Dependencies Rationale

The following table demonstrates the dependencies among the claimed security requirements. It shows that all dependencies are satisfied. Therefore the requirements work together to accomplish the overall objectives defined for the TOE.

Table 13 Requirement Dependencies

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1/Audit	FPT_STM.1	FPT_STM_EXT.1 included (which is hierarchic to FPT_STM.1)
FAU_GEN.1/IPS	FPT_STM.1	FPT_STM_EXT.1 included (which is hierarchic to FPT_STM.1)
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	FAU_GEN.1/Audit FIA_UIA_EXT.1, which specifies the relevant Administrator identification timing
FAU_STG.1/Audit	FAU_GEN.1	FAU_GEN.1/Audit
FAU_STG.1/IPS	FAU_GEN.1	FAU_GEN.1/IPS
FAU_STG_EXT.1	FAU_GEN.1, and FTP_ITC.1	FAU_GEN.1/Audit, FAU_GEN.1/IPS, and FTP_ITC.1
FAU_STG_EXT.3	FAU_GEN.1 FAU_STG_EXT.1	FAU_GEN.1/Audit, FAU_GEN.1/IPS, FAU_STG_EXT.1
FCS_CKM.1	(FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4	FCS_CKM.2 and FCS_CKM.4
FCS_CKM.2	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_CKM.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1
FCS_COP.1/DataEncryption	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1/SigGen	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1/Hash	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_COP.1/KeyedHash	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM.4
FCS_RBG_EXT.1	None	None - See DRBG Note Below.
FCS_SHC_EXT.1	FCS_CKM.1, FCS_CKM.2 FCS_COP.1/DataEncryption FCS_COP.1/SigGen FCS_COP.1/Hash FCS_COP.1/KeyedHash FCS_RBG_EXT.1	FCS_CKM.1, FCS_CKM.2 FCS_COP.1/DataEncryption FCS_COP.1/SigGen FCS_COP.1/Hash FCS_COP.1/KeyedHash FCS_RBG_EXT.1

ST Requirement	CC Dependencies	ST Dependencies
FCS_SHS_EXT.1	FCS_CKM.1, FCS_CKM.2 FCS_COP.1/DataEncryption FCS_COP.1/SigGen FCS_COP.1/Hash FCS_COP.1/KeyedHash FCS_RBG_EXT.1	FCS_CKM.1, FCS_CKM.2 FCS_COP.1/DataEncryption FCS_COP.1/SigGen FCS_COP.1/Hash FCS_COP.1/KeyedHash FCS_RBG_EXT.1
FIA_AFL.1	FIA_UAU.1	Satisfied by FIA_UIA_EXT.1, which specifies the relevant Administrator authentication
FIA_PMG_EXT.1	None	None
FIA_UAU_EXT.2	None	None
FIA_UIA_EXT.1	FTA_TAB.1	FTA_TAB.1
FIA_UAU.7	FIA_UAU.1	Satisfied by FIA_UIA_EXT.1, which specifies the relevant Administrator authentication
FMT_MOF.1/ManualUpdate	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MOF.1/Functions	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_SMF.1/Core	None	None
FMT_SMF.1/IPS	None	None
FMT_SMR.2	FIA_UID.1	Satisfied by FIA_UIA_EXT.1, which specifies the relevant Administrator identification
FPT_APW_EXT.1	None	None
FPT_SKP_EXT.1	None	None
FPT_STM_EXT.1	None	None
FPT_TST_EXT.1	None	None
FPT_TUD_EXT.1	FCS_COP.1/SigGen or FCS_COP.1/Hash	FCS_COP.1/SigGen
FTA_SSL.3	None	None
FTA_SSL.4	None	None
FTA_TAB.1	None	None
FTP_ITC.1	None	None
FTP_TRP.1	None	None
IPS_ABD_EXT.1	IPS_NTA_EXT.1 IPS_SBD_EXT.1	IPS_NTA_EXT.1 IPS_SBD_EXT.1
IPS_IPB_EXT.1	IPS_NTA_EXT.1 FMT_SMR.1	IPS_NTA_EXT.1 FMT_SMR.1
IPS_NTA_EXT.1	None	None
IPS_SBD_EXT.1	IPS_NTA_EXT.1	IPS_NTA_EXT.1
ADV_ARC.1	ADV_FSP.1, ADV_TDS.1	ADV_FSP.2, ADV_TDS.1
ADV_FSP.2	ADV_TDS.1	ADV_TDS.1
ADV_TDS.1	ADV_FSP.2	ADV_FSP.2

ST Requirement	CC Dependencies	ST Dependencies
AGD_OPE.1	ADV_FSP.1	ADV_FSP.2
AGD_PRE.1	None	None
ALC_CMC.2	ALC_CMS.1	ALC_CMS.2
ALC_CMS.2	None	None
ALC_DEL.1	None	None
ASE_CCL.1	ASE_INT.1, ASE_ECD.1, ASE_REQ.1	ASE_INT.1, ASE_ECD.1, ASE_REQ.2
ASE_ECD.1	None	None
ASE_INT.1	None	None
ASE_OBJ.2	ASE_SPD.1	ASE_SPD.1
ASE_REQ.2	ASE_ECD.1, ASE_OBJ.2	ASE_ECD.1, ASE_OBJ.2
ASE_SPD.1	None	None
ASE_TSS.1	ADV_FSP.1, ASE_INT.1, ASE_REQ.1	ADV_FSP.2, ASE_INT.1, ASE_REQ.2
ATE_COV.1	ADV_FSP.2 and ATE_FUN.1	ADV_FSP.2 and ATE_FUN.1
ATE_FUN.1	ATE_COV.1	ATE_COV.1
ATE_IND.2	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1
AVA_VAN.2	ADV_ARC.1 and ADV_FSP.2 and ADV_TDS.1 and AGD_OPE.1 and AGD_PRE.1	ADV_ARC.1 and ADV_FSP.2 and ADV_TDS.1 and AGD_OPE.1 and AGD_PRE.1

7.4 TOE Summary Specification

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This section, in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. Table 144 **Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

Table 144 Security Functions vs. Requirements Mapping

	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE Access	Trusted Path/Channels	Intrusion Prevention
FAU_GEN.1/Audit	X							
FAU_GEN.1/IPS	X							
FAU_GEN.2	X							
FAU_STG.1/Audit	X							
FAU_STG.1/IPS	X							
FAU_STG_EXT.1	X							
FAU_STG_EXT.3	X							
FCS_CKM.1		X						
FCS_CKM.2		X						
FCS_CKM.4		X						
FCS_COP.1/DataEncryption		X						
FCS_COP.1/SigGen		X						
FCS_COP.1/Hash		X						
FCS_COP.1/Keyed Hash		X						
FCS_RBG_EXT.1		X						
FCS_SHC_EXT.1		X						

	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE Access	Trusted Path/Channels	Intrusion Prevention
FCS_SHS_EXT.1		X						
FIA_AFL.1			X					
FIA_PMG_EXT.1			X					
FIA_UIA_EXT.1			X					
FIA_UAU_EXT.2			X					
FIA_UAU.7			X					
FMT_MOF.1/ManualUpdate				X				
FMT_MOF.1/Functions				X				
FMT_MTD.1				X				
FMT_SMF.1/Core				X				
FMT_SMF.1/IPS				X				
FMT_SMR.2				X				
FPT_APW_EXT.1					X			
FPT_SKP_EXT.1					X			
FPT_STM_EXT.1					X			
FPT_TST_EXT.1					X			
FPT_TUD_EXT.1					X			
FTA_SSL.3						X		
FTA_SSL.4						X		
FTA_TAB.1						X		
FTP_ITC.1							X	
FTP_TRP.1							X	
IPS_ABD_EXT.1								X
IPS_IPB_EXT.1								X
IPS_NTA_EXT.1								X

	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE Access	Trusted Path/Channels	Intrusion Prevention
IPS_SBD_EXT.1								X