Lancope, Inc.

StealthWatch v6.3.5

Security Target

Document Version: 1.3

Prepared for:

Lancope, Inc.
3650 Brookside Parkway, Suite 400
Alpharetta, GA 30022
United States of America

Phone: +1 (770) 225-6529
http://www.lancope.com

Prepared by:

Corsec Security, Inc.
13135 Lee Jackson Memorial Hwy., Suite 220
Fairfax, VA 22033
United States of America

Phone: +1 (703) 267-6050
http://www.corsec.com

Table of Contents

Table of Figures

List of Tables

1 Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the organization of the ST. The TOE is Lancope StealthWatch v6.3.5, and will hereafter be referred to as the TOE throughout this document. The TOE is a suite of network monitoring devices. These devices can be used in a variety of deployment configurations to gather a snapshot of network traffic and determine overall network health and performance. The devices can also analyze Netflow and various spinoff protocols in order to detect flow-based anomalies in network traffic.

1.1 Purpose

This ST is divided into nine sections, as follows:

- Introduction (Section 1) – Provides a brief summary of the ST contents and describes the organization of other sections within this document. It also provides an overview of the TOE security functions and describes the physical and logical scope for the TOE, as well as the ST and TOE references.
- Conformance Claims (Section 2) – Provides the identification of any Common Criteria (CC), Protection Profile, and Evaluation Assurance Level (EAL) package claims. It also identifies whether the ST contains extended security requirements.
- Security Problem (Section 3) – Describes the threats, organizational security policies, and assumptions that pertain to the TOE and its environment.
- Security Objectives (Section 4) – Identifies the security objectives that are satisfied by the TOE and its environment.
- Extended Components (Section 5) – Identifies new components (extended Security Functional Requirements (SFRs) and extended Security Assurance Requirements (SARs)) that are not included in CC Part 2 or CC Part 3.
- Security Requirements (Section 6) – Presents the SFRs and SARs met by the TOE.
- TOE Summary Specification (Section 7) – Describes the security functions provided by the TOE that satisfy the security functional requirements and objectives.
- Rationale (Section 8) - Presents the rationale for the SFR dependencies as to their consistency, completeness, and suitability.
- Acronyms and Terms (Section 9) – Defines the acronyms and terminology used within this ST.

1.2 Security Target and TOE References

Table 1 below shows the ST and TOE references.

Table 1 ST and TOE References

ST Title	Lancope, Inc. StealthWatch v6.3.5 Security Target
ST Version	Version 1.3
ST Author	Corsec Security, Inc.
ST Publication Date	2014-04-03
TOE Reference	Lancope StealthWatch v6.3.5
FIPS[1] 140-2 Status	Level 1, Validated crypto module, Certificate No. 1291

[1] FIPS – Federal Information Processing Standard

1.3 Product Overview

The StealthWatch system is a scalable network monitoring system. StealthWatch continuously monitors network traffic for health, performance, and security anomalies. Once the system is installed on a network, it begins gathering data from:

- flows
- hosts
- services
- applications

1.3.1 Flows

Flows are a summary of all packet data traveling across the network (usually including header information at multiple levels, but not actual payload data). Flows tend to persist for the duration of a session (that is, from the initiation of a protocol handshake until the termination of that same session). Flows also include bidirectional data, that is data flowing from the source to the destination, and the data being sent back from the destination to the source. By examining flows instead of all packet information, StealthWatch uses fewer resources to diagnose and resolve network issues.

Flows contain:

- input and output interface indices,
- timestamps for the flow start and finish time in milliseconds since the last hour,
- number of bytes and packets observed in the flow,
- layer 3 headers (source and destination IP[2] addresses and port numbers, IP protocol, and type of service values)
- for Transmission Control Protocol (TCP) flows, the union of all TCP flags observed over the life of the flow,
- layer 3 routing information (IP address of the immediate next hop to the destination and source and destination IP masks).

If NetFlow v9 is supported on the network, StealthWatch can additionally monitor Multiprotocol Label Switching (MPLS)[3] labels and IPv6 addresses and ports.

Since NetFlow and similar protocols provide only unidirectional flow data, StealthWatch combines multiple unidirectional flows into bidirectional flows via a process called "flow stitching". The net effect of collecting this data is that StealthWatch can create a baseline profile for expected network behavior from each host. When network activity deviates from this baseline, StealthWatch detects the anomalies and can either alert administrators or take other actions as configured by an administrator.

StealthWatch is able to deduplicate flows as they are received, preventing any flows that might have traversed multiple routers from creating duplicate data trails. This prevents misreporting the volume of attack traffic and does not force users to select a specific router when analyzing traffic between hosts. StealthWatch attempts to optimize this process by selecting the ideal source for flow data when deduplicating data flows. During deduplication, StealthWatch does not discard data from multiple flows, but instead maintains a single deduplicated count for bytes, packets, etc.

Deep Packet Inspection (DPI) is performed on packets to determine the application layer protocols being used. DPI is the technique of looking within a packet's payload at the data and nested headers of higher

[2] IP – Internet Protocol

[3] MPLS is a protocol that allows networks to direct data from one node to the next by using short path labels instead of long network addresses, which results in fewer resource-intensive lookups in routing tables.

layer protocols in order to determine information about the packet. This is how StealthWatch obtains application level information about data traveling across the network. Although DPI is performed, StealthWatch only accesses the nested header data in order to provide comprehensive information about the data traveling across the network. This allows DPI to enhance the functionality of the StealthWatch system without the typical performance reduction associated with DPI functionality.

In addition to the above types of data, StealthWatch can also collect syslog messages from firewalls, intrusion detection systems, and intrusion prevention systems. These are classified as external events and associated with flow-driven events.

1.3.2 Hosts

Hosts include computers, routers, switches, or any other device on the network that has an IP address and is creating TCP/IP traffic (including virtual devices). StealthWatch monitors traffic transmitted by each host. These are the devices that originate the data for StealthWatch to analyze.

1.3.3 Services

Services are IP packets that use specific TCP or User Datagram Protocol (UDP) ports and allow hosts to access other hosts or servers. An example of a service is Hypertext Transfer Protocol (HTTP), which allows clients to use port 80 to communicate with web servers. StealthWatch comes preconfigured to recognize many of the most common services, and provides the capacity for administrators to define more as needed. Even if a service has not been defined, StealthWatch still gathers data on the service if it appears on the network.

1.3.4 Applications

Applications are similar to services in that they are a combination of IP packets and TCP or UDP ports. Applications are different from services in that the specific source and destination hosts and servers involved in the data flows are also identified and tracked. This allows StealthWatch to determine not only the type of traffic, but to classify traffic depending on its destination (e.g. Facebook traffic is differentiated from SalesForce traffic).

1.3.5 Product Components

The StealthWatch v6.3.5 functionality is divided among three distinct devices:

- FlowCollector models 1000, 2000, and 4000
- FlowSensor models 1000, 2000, and 3000
- StealthWatch Management Console (SMC) models 1000 and 2000

1.3.5.1 FlowCollector

The FlowCollector device serves as the network flow collection and analysis point for the StealthWatch system. FlowCollector receives NetFlow, cFlow, J-Flow, Packeteer 2, NetStream, IPFIX[4], sFlow, Syslog, and SNMP[5] data directly from network taps or Switched Port Analyzer (SPAN) ports, or from FlowSensor Appliances. All network flow data acquired by the StealthWatch system is aggregated by the FlowCollector.

[4] IPFIX – Internet Protocol Flow Information Export
[5] SNMP – Simple Network Management Protocol

1.3.5.2 FlowSensor

The FlowSensor and FlowSensor VE[6] gather packet-level details of network flows in order to provide DPI of network data flows. This allows the FlowSensor to gather application- and performance-specific information on packets across the network. FlowSensor is installed in areas of the network that do not support NetFlow (or similar protocols) or in areas where more information is desired about data traveling across the network than NetFlow can provide. This allows FlowSensor to fill in gaps that FlowCollector would otherwise miss. Network data gathered by the FlowSensor is exported to the FlowCollector as NetFlow v9 records. FlowSensor VE performs the same tasks as a FlowSensor but in a virtual instead of physical environment.

1.3.5.3 SMC

The Management Console manages the other devices and provides the user interfaces that allow Administrators to control the configuration for each StealthWatch device. Other devices forward flow data to the Management Console for analysis and reporting. Administrators can review gathered data from the SMC web interface.

1.4 TOE Overview

The TOE Overview provides a high-level description of the Lancope StealthWatch v6.3.5 that is the subject of the evaluation. The following section, TOE Overview, provides the introduction to the parts of the overall product offering that are specifically being evaluated.

The TOE is a distributed network monitoring system that has numerous security management features. The TOE is a hardware TOE that includes StealthWatch v6.3.5 running on a FlowCollector, FlowSensor, and SMC. Each component can be deployed on the appliances listed above, which differ in capacity, performance, and scalability options, but maintain the same security functionality. The TOE includes all of the components listed above and the security functionality listed in 1.5.2. The scope of this evaluation was on the secure management of the TOE. Section 1.4.2 identifies any major non-TOE hardware and software that is required by the TOE.

1.4.1 StealthWatch Interfaces

Each StealthWatch component uses a curses-based Command Line Interface (CLI) for maintenance and configuring hosts settings (IP address, DNS[7], etc.) and a Web User Interface (UI) for audit and cryptographic administration. There is also a web-based Graphical User Interface (GUI) on the SMC device that is also called the SMC. This GUI provides all of the configuration, data analysis, reporting, network monitoring, and other administrative functionality. SMC also provides a Web Services Application Programming Interface (API) that allows third-party systems to integrate with the SMC to gather data using Simple Object Access Protocol (SOAP). Additionally, the each component contains a syslog interface for exporting audit records. All StealthWatch components use HTTPS[8] tunnels to securely communicate management data to each other.

1.4.2 TOE Environment

The StealthWatch Management Console is a thick-client GUI that mirrors the functionality of the web interface and is a required environmental component. The SMC requires a general purpose computer with Java v5 (or later) and either Firefox 3.0 (or later) or Internet Explorer 8.0 (or later). SMC also relies on the network infrastructure to connect the general purpose computer to the SMC device. The CLI on each appliance also requires a general purpose computer (this can be the same as the one accessing the SMC).

The environment must contain a syslog server connected to the network where the TOE is installed.

[6] VE – Virtual Edition
[7] DNS – Domain Name Service
[8] HTTPS – Hypertext Transfer Protocol Secure

It is assumed that only trusted users or software have access to the TOE hardware components. In addition, the TOE hardware components are intended to be deployed in a physically secured cabinet, room, or data center with the appropriate level of physical access control and physical protection (e.g. badge access, fire control, locks, alarms, etc.).

The FlowCollector requires flow exporting devices such as routers, firewalls, and switches. The FlowSensor requires a switch port analyzer (SPAN), mirror port, or Ethernet test access port (TAP). See Section 1.5.1 below for a detailed description of the environment relied upon by the TOE components.

1.5 TOE Description

This section primarily addresses the physical and logical components of the TOE included in the evaluation.

1.5.1 Physical Scope

The physical scope of the TOE includes the TOE software installed on top of a hardware appliance. For the evaluated configuration of the TOE, the software and hardware are StealthWatch v6.3.5 running on FlowCollector 1000, 2000, or 4000; FlowSensor 1000, 2000, or 3000; and StealthWatch Management Console 1000 or 2000. The FlowCollector, StealthWatch Management Console, and FlowSensor components are hardware, while the StealthWatch component is software. The StealthWatch software runs on all appliances and is derived from a single image, with different functionality enabled or disabled based on the hardware it is installed on. All appliances run the TOE software that differs only in platform-specific configuration data, which describe the intended hardware platform to the operating system. Differences between product models allow for different capacity, performance, and scalability options. All platforms use Intel Xeon processors.

Figure 1 illustrates the physical scope and the physical boundary of the overall solution and ties together all of the components of the TOE and the constituents of the TOE Environment. The TOE is composed of custom hardware and software running on multiple devices on a network. Although the hardware is different for each device, the software is provided by the same code with different portions activated, depending on the hardware platform where the software is installed. The TOE Components are the same as the product components as specified in section 1.3.5.

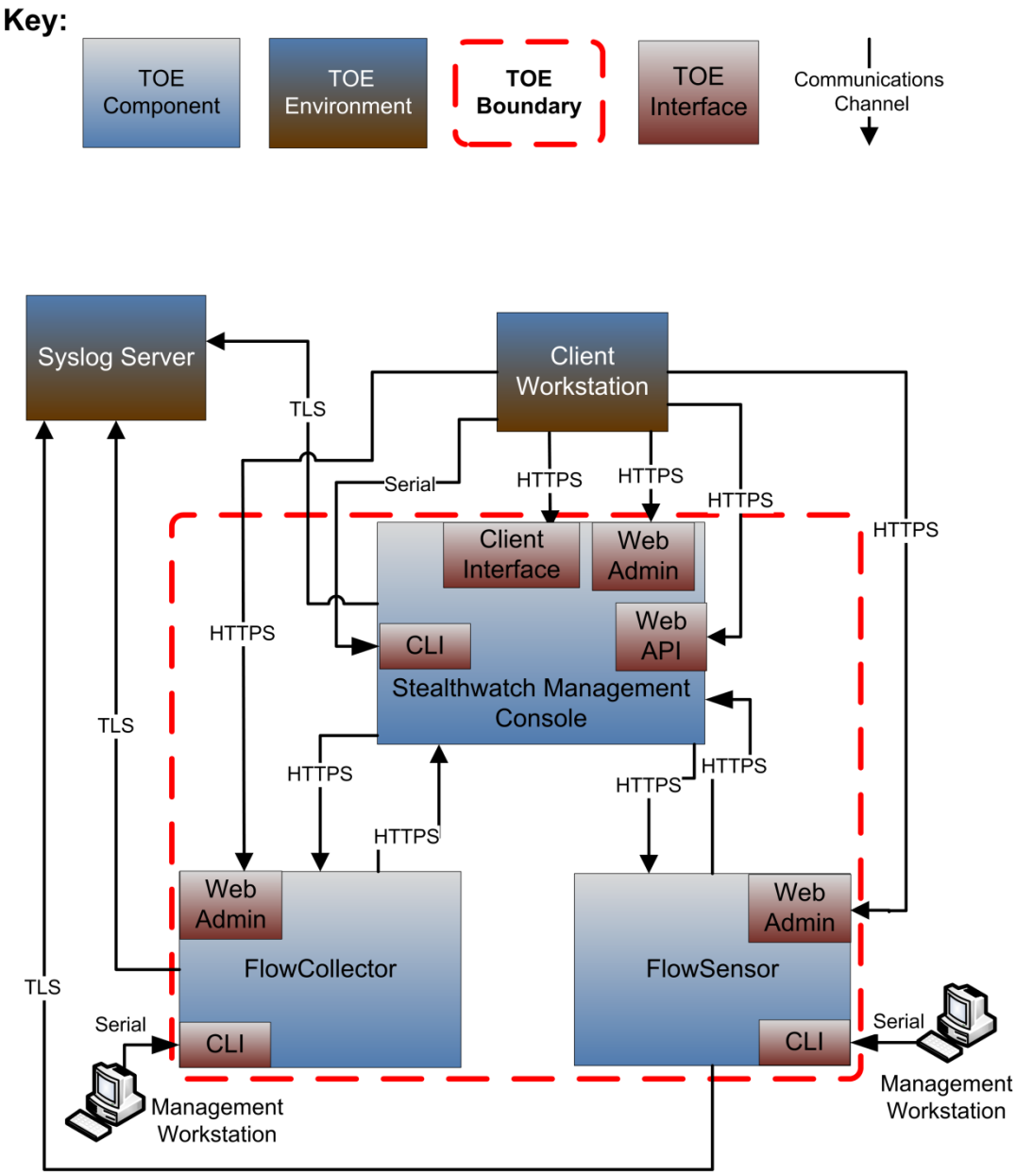

Figure 1 Physical TOE Boundary

The TOE Boundary includes all the Lancope developed hardware and software components of the StealthWatch v6.3.5 product. Any third party source code or software that StealthWatch v6.3.5 has modified is considered to be TOE Software.

1.5.1.1 Guidance Documentation

Table 2 lists the TOE Guidance Documentation to install, configure, and maintain the TOE.

Table 2 Guidance Documentation

Document Name	Description
StealthWatch System Hardware	Includes steps for installing the TOE hardware into a server rack

Document Name	Description
Installation Guide, Version v6.3	and initialize the management interfaces to receive connections.
StealthWatch System Hardware Configuration Guide For v6.3	Includes steps for the initial configuration of the TOE software.
StealthWatch Management Console User's Guide For v6.3	Contains explanations of each of the screens and basic operations available via the SMC.
StealthWatch Management Console Admin Interface Online Help for v6.3	Online help file that is accessible through the SMC.
StealthWatch Management Console Web Interface Online Help for v6.3	Online help file that is accessible through the Web UI.
StealthWatch Management Console Client Interface Help File for v6.3	Online help file that is accessible through the thick client.
SMC Web Services Programming Guide for SMC v6.3	A guide for using the Web Services API.
StealthWatch What's New in StealthWatch System v6.3	Includes details of the latest release.
Guidance Supplement v0.1	Contains information regarding specific configuration for the TOE evaluated configuration.
StealthWatch System Version 6.3.X Update Guide	Provides the steps to perform a system update.

1.5.2 Logical Scope

The logical boundary of the TOE will be broken down into the following security classes which are further described in sections 6 and 7 of this ST. The logical scope also provides the description of the security features of the TOE. The security functional requirements implemented by the TOE are usefully grouped under the following Security Function Classes:

1.5.2.1 Security Audit

The TOE generates audit records for security relevant actions of the authorized administrators within the SMC. The TOE provides an authorized administrator access to view the audit logs created as a result of administrator actions through the SMC. The TOE records the identity of the User responsible for the log event, where applicable. All logs are backed up to a syslog server via a secure channel.

1.5.2.2 Cryptographic Support

The Cryptographic Support TSF[9] provides cryptographic functions to secure communications for SMC management sessions and between physically separate TOE Components. TLS and HTTPS are used to secure these communications sessions. In addition, the TOE provides a variety of cryptographic algorithms for its own use.

1.5.2.3 User Data Protection

The TOE stores network data within volatile memory while the data is being used by the TOE. Once the TOE finishes using the packet data, or if the TOE is rebooted, the memory space is de-allocated and zeroized.

[9] TSF – TOE Security Functionality

1.5.2.4 Identification and Authentication

The TOE provides functionality that requires administrators to verify their claimed identity. The Identification and Authentication TSF[10] ensures that only legitimate administrators can gain access to the configuration settings and management settings of the TOE. Administrators must log in with a valid user name and password before the TOE will permit the administrators to manage the TOE. The TOE requires administrators to use strong passwords. No feedback is presented to Users when they are entering their passwords at the login prompt of the CLI, while obscured feedback is presented to Users when they are entering their passwords at the login prompt of the web interface.

1.5.2.5 Security Management

The TOE provides a feature-rich GUI called the SMC for administrators to manage the security functions, configuration, and other features of the TOE components. The Security Management function specifies user roles with defined access for the management of the TOE components. Additionally, the TOE provides a CLI that Administrators can use to perform maintenance tasks on the TOE.

1.5.2.6 Protection of the TSF

The TOE implements HTTPS for protection of the SMC. HTTPS (TLS) connections are used to protect all communication between the TOE and SMC. HTTPS uses the TOE's cryptographic capabilities to protect communications.

The management communication channels between the TOE and remote entities are distinct from other communication channels and provide assured identification of both endpoints. In addition, the communications are protected from modification and disclosure.

Cryptographic keys are protected from being read by external entities since they are only accessible to the cryptographic code on the TOE. At startup, the TOE runs a suite of self-tests that verify the correct operation of all cryptographic code.

The TOE also provides a reliable timestamp for its own use. A digital signature is used to verify all software updates that are applied to the TOE.

1.5.2.7 TOE Access

The TOE terminates local and remote management sessions after an administrator-configurable time period of inactivity. After a User's session is terminated, the user must log in again to regain access to TOE functionality. A login banner is displayed for users at the login screen of the SMC GUI and at the login prompt of the CLI.

1.5.2.8 Trusted Path/Channels

The TOE implements a trusted TLS tunnel between itself and a remote syslog server in order to protect syslog traffic as it is being sent to the server. Additionally, the TOE provides trusted paths between Administrators and the SMC GUI via an HTTPS tunnel. All tunnels are encrypted.

1.5.3 Scope of Evaluation

The evaluation is limited in scope to the secure management features described in Network Device Protection Profile (NDPP) v1.1 and detailed in section 1.5.2.

[10] TOE Security Functionality

2 Conformance Claims

This section provides the identification for any CC, Protection Profile (PP), and EAL package conformance claims. Rationale is provided for any extensions or augmentations to the conformance claims. Rationale for CC and PP conformance claims can be found in Section 8.1.

Table 3 CC and PP Conformance

Common Criteria (CC) Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012; CC Part 2 extended; CC Part 3 conformant; PP claim Network Devices Protection Profile conformant; Parts 2 and 3 Interpretations of the CEM[11] as of 2012-05-25 were reviewed, and no interpretations apply to the claims made in this ST.
PP Identification	Exact Conformance[12] to Network Devices Protection Profile v1.1

[11] Common Evaluation Methodology

[12] Exact Conformance is a type of Strict Conformance such that the set of SFRs and the SPD/Objectives are exactly as presented within the accepted NDPP without changes.

3 Security Problem

This section describes the security aspects of the environment in which the TOE will be used and the manner in which the TOE is expected to be employed. It provides the statement of the TOE security environment, which identifies and explains all:

- Known and presumed threats countered by either the TOE or by the security environment
- Organizational security policies with which the TOE must comply
- Assumptions about the secure usage of the TOE, including physical, personnel and connectivity aspects

3.1 Threats to Security

This section identifies the threats to the Information Technology (IT) assets against which protection is required by the TOE or by the security environment. The threat agents are divided into two categories:

- Attackers who are not TOE users: They have public knowledge of how the TOE operates and are assumed to possess a low skill level, limited resources to alter TOE configuration settings or parameters and no physical access to the TOE.
- TOE users: They have extensive knowledge of how the TOE operates and are assumed to possess a high skill level, moderate resources to alter TOE configuration settings or parameters and physical access to the TOE. (TOE users are, however, assumed not to be willfully hostile to the TOE.)

Both are assumed to have a low level of motivation. The IT assets requiring protection are the TSF and user data saved on or transitioning through the TOE and the hosts on the protected network. Removal, diminution, and mitigation of the threats are through the objectives identified in Section 4 Security Objectives. Table 4 below lists the applicable threats.

Table 4 Threats

Name	Description
T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNDETECTED_ACTIONS	Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender.

3.2 Organizational Security Policies

An Organizational Security Policy (OSP) is a set of security rules, procedures, or guidelines imposed by an organization on the operational environment of the TOE. Table 5 below lists the OSPs that are presumed to be imposed upon the TOE or its operational environment by any organization implementing the TOE in the CC evaluated configuration.

Table 5 Organizational Security Policies

Name	Description
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.

3.3 Assumptions

This section describes the security aspects of the intended environment for the evaluated TOE. The operational environment must be managed in accordance with assurance requirement documentation for delivery, operation, and user guidance. Table 6 lists the specific conditions that are required to ensure the security of the TOE and are assumed to exist in an environment where this TOE is employed.

Table 6 Assumptions

Name	Description
A.NO_GENERAL_PURPOSE	It is assumed that 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	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

4 Security Objectives

Security objectives are concise, abstract statements of the intended solution to the problem defined by the security problem definition (see Section 3). The set of security objectives for a TOE form a high-level solution to the security problem. This high-level solution is divided into two part-wise solutions: the security objectives for the TOE, and the security objectives for the TOE's operational environment. This section identifies the security objectives for the TOE and its supporting environment.

4.1 Security Objectives for the TOE

The specific security objectives for the TOE are as follows:

Table 7 Security Objectives for the TOE

Name	Description
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
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 (optionally) from a trusted source.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity.
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 configure the TOE, and provide protections for logged-in administrators.
O.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.SESSION_LOCK	The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

4.2 Security Objectives for the Operational Environment

4.2.1 IT Security Objectives

The following IT security objectives are to be satisfied by the environment:

Table 8 IT Security Objectives

Name	Description
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers

Name	Description
	or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

4.2.2 Non-IT Security Objectives

The following non-IT environment security objectives are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Table 9 Non-IT Security Objectives

Name	Description
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

5 Extended Components

This section defines the extended SFRs and extended SARs met by the TOE. These requirements are presented following the conventions identified in Section 6.1.

5.1 Extended TOE Security Functional Components

This section specifies the extended SFRs for the TOE. The extended SFRs are organized by class. Table 10 identifies all extended SFRs implemented by the TOE.

Table 10 Extended TOE Security Functional Requirements

Name	Description
FAU_STG_EXT.1	External Audit Trail Storage
FCS_CKM_EXT.4	Cryptographic key destruction
FCS_HTTPS_EXT.1	Explicit: HTTPS
FCS_RBG_EXT.1	Extended: Cryptographic Operation (Random Bit Generation)
FCS_TLS_EXT.1	Explicit: TLS
FIA_PMG_EXT.1	Password Management
FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism
FIA_UIA_EXT.1	User Identification and Authentication
FPT_APW_EXT.1	Extended: Protection of Administrator Passwords
FPT_SKP_EXT.1	Extended: Protection of TSF data (for reading of all symmetric keys)
FPT_TST_EXT.1	TSF self test
FPT_TUD_EXT.1	Extended: Trusted Update
FTA_SSL_EXT.1	TSF-initiated session locking

5.1.1 Class FAU: Security Audit

Families in this class address the requirements for functions to implement security audit as defined in CC Part 2.

5.1.1.1 Family FAU_STG: Security audit event storage

Family Behaviour

This family defines the requirements for the TSF to be able to create and maintain a secure audit trail. Stored audit records refers to those records within the audit trail, and not the audit records that have been retrieved (to temporary storage) through selection.

Components in this family address the requirements for protection audit data as defined in CC Part 2. This section defines the extended components for the FAU_STG family.

Component Leveling

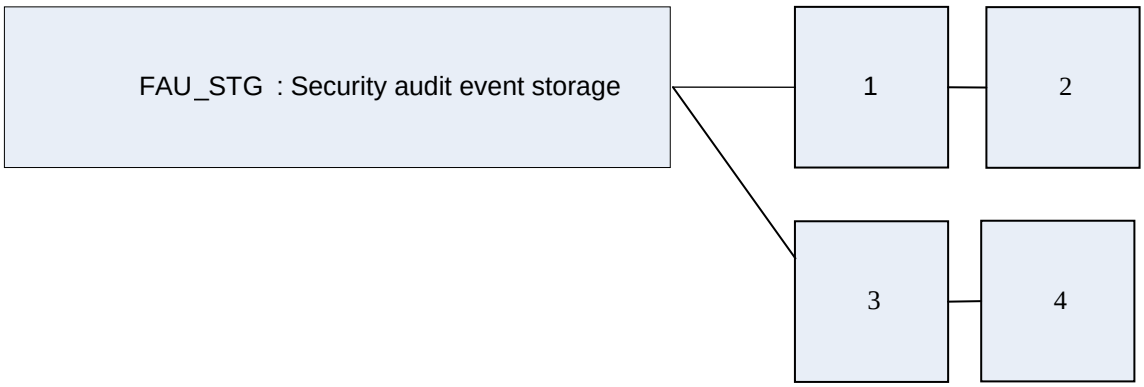

Figure 2 Security audit event storage family decomposition

The extended FAU_STG_EXT.1 component is considered to be part of the FAU_STG family.

FAU_STG_EXT.1 External Audit Trail Storage requires the TSF to use an external IT entity for audit data storage. It was modeled after FAU_STG.1.

Management: FAU_STG_EXT.1

a) There are no management activities foreseen.

Audit: FAU_STG_EXT.1

a) There are no audit activities foreseen.

FAU_STG_EXT.1 External Audit Trail 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 [selection: transmit the generated audit data to an external IT entity, receive and store audit data from an external IT entity] using a trusted channel implementing the [selection: IPsec, SSH, TLS, TLS/HTTPS] protocol.

5.1.2 Class FCS: Cryptographic Support

Families in this class address the requirements for functions to implement cryptographic functionality as defined in CC Part 2.

5.1.2.1 Family FCS_CKM: Cryptographic Key Management

Family Behaviour

Cryptographic keys must be managed throughout their life cycle. This family is intended to support that lifecycle and consequently defines requirements for the following activities: cryptographic key generation, cryptographic key distribution, cryptographic key access and cryptographic key destruction. This family should be included whenever there are functional requirements for the management of cryptographic keys.

Components in this family address the requirements for managing cryptographic keys as defined in CC Part 2. This section defines the extended components for the FCS_CKM family.

Component Leveling

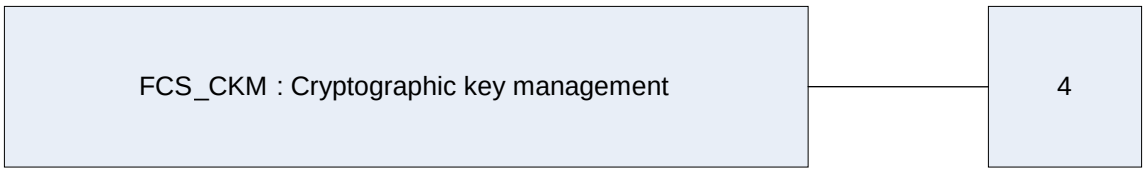

Figure 3 Cryptographic key management family decomposition

The extended FCS_CKM_EXT.4 component is considered to be part of the FCS_CKM family.

FCS_CKM_EXT.4 Cryptographic key zeroization, requires cryptographic keys and cryptographic critical security parameters to be zeroized. It was modeled after FCS_CKM.1

Management: FCS_CKM_EXT.4

a) There are no management activities foreseen.

Audit: FCS_CKM_EXT.4

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

a) Failure on invoking the cryptographic key zeroization functionality.

FCS_CKM_EXT.4 Cryptographic Key Zeroization
Hierarchical to: FCS_CKM.4
Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM_EXT.4.1

The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs[13] when no longer required.

[13] Critical Security Parameter

5.1.2.2 Family FCS_HTTPS_EXT: Extended: HTTPS

Family Behaviour

Components in this family address the requirements for protecting communications using HTTPS. This is a new family defined for the FCS Class.

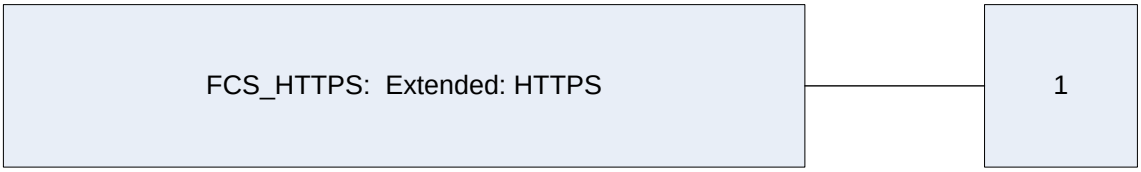

Figure 4 Extended: HTTPS family decomposition

FCS_HTTPS_EXT.1 Extended: HTTPS, requires that HTTPS be implemented.

Management: FCS_HTTPS_EXT.1

a) There are no management activities foreseen.

Audit: FCS_ HTTPS _EXT.1

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

a) There are no auditable events foreseen.

FCS_HTTPS_EXT.1 Extended: HTTPS
Hierarchical to: No other components
Dependencies: FCS_TLS_EXT.1 Extended: TLS
FCS_HTTPS_EXT.1.1
The TSF shall implement the HTTPS protocol that complies with RFC[14] 2818.
FCS_HTTPS_EXT.1.2
The TSF shall implement the HTTPS protocol using TLS as specified in FCS_TLS_EXT.1.

[14] RFC – Request For Comments

5.1.2.3 Family FCS_RBG_EXT: Extended: Random Bit Generation

Family Behaviour

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

Component Leveling

Figure 5 Extended: Random Bit Generation family decomposition

FCS_RBG_EXT.1 Extended: Random Bit Generation, requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source. It was modeled after FCS_COP.1

Management: FCS_RBG_EXT.1

a) There are no management activities foreseen.

Audit: FCS_RBG_EXT.1

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

a) Failure of the randomization process.

FCS_RBG_EXT.1 Extended: Random Bit Generation
Hierarchical to: No other components
Dependencies: None.
FCS_RBG_EXT.1.1

The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: NIST[15] Special Publication 800-90 using [selection: Hash_DRBG[16] (any), HMAC[17] DRBG (any), CTR[18] DRBG (AES[19]), Dual EC[20] DRBG (any)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulated entropy from [selection, one or both of: a software-based noise source; a TSF-hardware-based noise source].

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded with a minimum of [selection, choose one of: 128 bits, 256 bits] of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

[15] NIST – National Institute of Standards and Technology
[16] DRBG – Deterministic Random Bit Generator
[17] HMAC – Hashed Message Authentication Code
[18] CTR – Counter Mode
[19] AES – Advanced Encryption Standard
[20] EC – Elliptical Curve

5.1.2.4 Family FCS_TLS_EXT: Extended: TLS

Family Behaviour

Components in this family address the requirements for protecting communications using TLS. This is a new family defined for the FCS Class.

Component Leveling

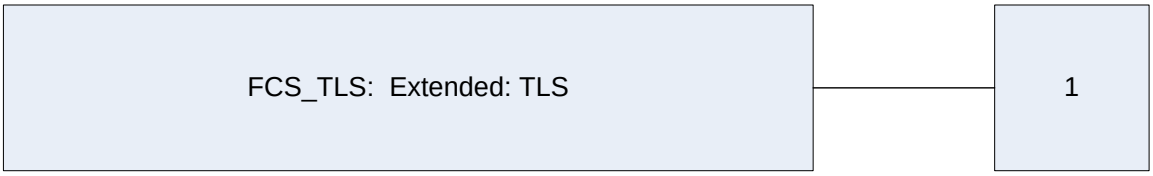

Figure 6 Extended: TLS family decomposition

FCS_TLS_EXT.1 Extended: TLS, requires that TLS be implemented.

Management: FCS_TLS_EXT.1

a) There are no management activities foreseen.

Audit: FCS_ TLS _EXT.1

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

a) There are no auditable events foreseen.

FCS_TLS_EXT.1 Extended: TLS
Hierarchical to: No other components
Dependencies: FCS_COP.1(1) Cryptographic operation (for data encryption/decryption)
FCS_TLS_EXT.1.1

The TSF shall implement one or more of the following protocols [selection: TLS 1.0 (RFC 2346), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA[21]
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA

[selection:
None
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256[22]
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
].

[21] CBC – Cipher Block Chaining, SHA – Secure Hash Algorithm
[22] ECDSA – Elliptical Curve Digital Signature Algorithm

5.1.3 Class FIA: Identification and Authentication

Families in this class address the requirements for functions to establish and verify a claimed user identity as defined in CC Part 2.

5.1.3.1 Family FIA_PMG_EXT: Password Management

Family Behaviour

This family defines the password strength rules enforced by the TSF.

This section defines the extended components for the FIA_PMG_EXT family.

Component Leveling

Figure 7 Extended: Password Management family decomposition

The extended FIA_PMG_EXT.1 component is considered to be part of the FIA_PMG_EXT family.

FIA_PMG_EXT.1 defines the password strength requirements that the TSF will enforce.

Management: FIA_PMG_EXT.1

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

a) Administrator configuration of strength requirements.

Audit: FIA_PMG_EXT.1

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

a) Changes to strength requirements.
b) Rejection of user password based on failure to comply with requirements.

FIA_PMG_EXT.1 Password Management
Hierarchical to: No other components.
FIA_PMG_EXT.1.1

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

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters: [selection: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: *other characters*]];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

Dependencies: No dependencies

5.1.3.2 Family FIA_UAU: User authentication

Family Behaviour

This family defines the types of user authentication mechanisms supported by the TSF.

This section defines the extended components for the FIA_UAU family.

Component Leveling

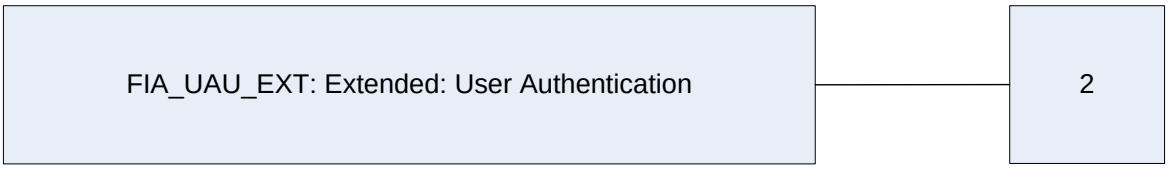

Figure 8 User Authentication family decomposition

The extended FIA_UAU_EXT.2 component is considered to be part of the FIA_UAU family as defined in CC Part 2.

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism, requires a local password-based authentication mechanism and the capability for passwords to expire. In addition, other authentication mechanisms can be specified.

Management: FIA_UAU_EXT.2

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

b) reset a user password by an administrator.

Audit: FIA_UAU_EXT.2

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

a) All use of the authentication mechanisms.

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism
Hierarchical to: No other components
Dependencies: No dependencies.
FIA_UAU_EXT.2.1

The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: *other authentication mechanism(s)*], none] to perform user authentication.

5.1.3.3 Family FIA_UIA_EXT: User identity and authentication

Family Behaviour

This family defines the types of user identification and authentication mechanisms supported by the TSF.

This section defines the components for the extended FIA_UIA_EXT family.

Component Leveling

Figure 9 Extended: User Identification and Authentication family decomposition

The extended FIA_UIA_EXT.1 component is considered to be part of the FIA_UIA_EXT family and is based on a combination of FIA_UAU.1 and FIA_UID.1.

FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism, requires a local password-based authentication mechanism and the capability for passwords to expire. In addition, other authentication mechanisms can be specified.

Management: FIA_UIA_EXT.1

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

a) Management of the authentication data by an administrator;
b) Management of the authentication data by the associated user;
c) Managing the list of actions that can be taken before the user is identified and authenticated;
d) Management of the user identities.

Audit: FIA_UIA_EXT.1

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

a) Minimal: Unsuccessful use of the identification or authentication mechanism, including the user identity provided;
b) Basic: All use of the identification and authentication mechanism, including the user identity provided;
c) Detailed: All TSF mediated actions performed before authentication of the user.

FIA_UIA_EXT.1 User Identification and Authentication

Hierarchical to: FIA_UID.1 Timing of identification
FIA_UAU.1 Timing of Authentication

FIA_UIA_EXT.1.1

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

- Display the warning banner in accordance with FTA_TAB.1;
- [selection: no other actions, [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.

Dependencies: No dependencies

5.1.4 Class FPT: Protection of the TSF

Families in this class address the requirements for functions providing integrity and management of mechanisms that constitute the TSF and of the TSF data as defined in CC Part 2.

5.1.4.1 Family FPT_APW: Extended: Protection of Administrator Passwords

Family Behaviour

Components in this family address the requirements for protection of administrator passwords. This is a new family defined for the FPT class.

Component Leveling

Figure 10 Extended: Protection of Administrator Passwords family decomposition

FPT_APW_EXT.1 Extended: Protection of Administrator Passwords, requires administrator passwords to be stored in non-plaintext form and requires the TOE to prevent reading of plaintext passwords. It was modeled after FPT_SSP.2.

Management: FPT_APW_EXT.1

a) There are no management activities foreseen.

Audit: FPT_APW_EXT.1

a) There are no audit activities foreseen.

FPT_APW_EXT.1 Extended: Protection of Administrator Passwords
Hierarchical to: No other components
Dependencies: None.
FPT_APW_EXT.1.1
The TSF shall store passwords in non-plaintext form.
FPT_APW_EXT.1.2
The TSF shall prevent the reading of plaintext passwords.

5.1.4.2 Family FPT_SKP: Extended: Protection of TSF data (for reading of all symmetric keys)

Family Behaviour

Components in this family address the requirements for protection of symmetric keys stored on the TOE.

Component Leveling

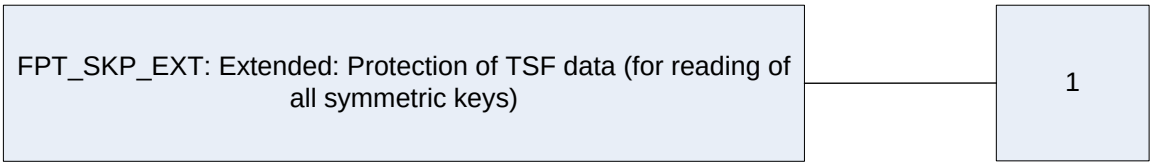

Figure 11 Extended: Protection of TSF data (for reading of all symmetric keys)

FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys), requires the TOE to prevent reading of all pre-shared, symmetric, and private keys. It was modeled after FPT_SSP.1.

Management: FPT_SKP_EXT.1

a) There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

a) There are no audit activities foreseen.

FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys)
Hierarchical to: No other components
Dependencies: No dependencies.
FPT_SKP_EXT.1.1
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.1.4.3 Family FPT_TST: TSF self test

Family Behaviour

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

The extended FPT_TST_EXT.1 component is considered to be part of the FPT_TST family.

Component Leveling

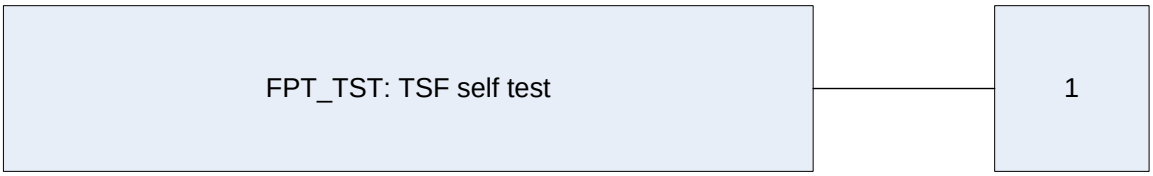

Figure 12 TSF testing family decomposition

FPT_TST _EXT.1 Extended: TSF testing, requires a suite of self tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

Management: FPT_TST _EXT.1

a) There are no management activities foreseen.

Audit: FPT_TST _EXT.1

a) There are no auditable activities foreseen.

FPT_TST_EXT.1 TSF testing
Hierarchical to: No other components.
Dependencies: No dependencies.
FPT_TST_EXT.1.1
The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

5.1.4.4 Family FPT_TUD: Extended: Trusted Update

Components in this family address the requirements for updating the TOE firmware and/or software. This is a new family defined for the FPT Class.

Component Leveling

Figure 13 Extended: Trusted Update family decomposition

FPT_TUD_EXT.1 Extended: Management of TSF Data, requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

Management: FPT_ TUD_EXT.1

a) There are no management activities foreseen.

Audit: FPT_ TUD_EXT.1

a) There are no auditable activities foreseen.

FPT_ TUD_EXT.1 Extended: Trusted Update
Hierarchical to: No other components.
Dependencies: FCS_COP.1(2) Cryptographic operation (for cryptographic signature).
FCS_COP.1(3) Cryptographic operation (for cryptographic hashing).

FPT_TUD_EXT.1.1
The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2
The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

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

5.1.5 Class FTA: TOE Access

Family Behaviour

Families in this class address the requirements for functions that control the establishment and existence of a user session as defined in CC Part 2.

5.1.5.1 Family FTA_SSL: TSF-initiated Session Locking

Components in this family address the requirements for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

The extended FTA_SSL_EXT.1 component is considered to be part of the FTA_SSL family.

Component Leveling

Figure 14 TSF-initiated Session Locking family decomposition

FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking, requires system initiated locking of an interactive session after a specified period of inactivity.

Management: FTA_SSL_EXT.1

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

a) Specification of the time of user inactivity after which lock-out occurs for an individual user.

Audit: FTA_SSL_EXT.1

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

a) Any attempts at unlocking an interactive session.

FTA_SSL_EXT.1 Extended: TSF self test
Hierarchical to: No other components.
Dependencies: FIA_UAU.1 Timing of authentication.
FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [selection:

- lock the session – disable any activity of the user's data access display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session;
- terminate the session]

after a Security Administrator-specified time period of inactivity.

5.2 Extended TOE Security Assurance Components

There are no extended TOE Security Assurance Components.

6 Security Requirements

This section defines the SFRs and SARs met by the TOE. These requirements are presented following the conventions identified in Section 6.1.

6.1 Conventions

There are several font variations used within this ST. Selected presentation choices are discussed here to aid the Security Target reader.

The CC allows for assignment, refinement, selection and iteration operations to be performed on security functional requirements. All of these operations are used within this ST. These operations are performed as described in Part 2 of the CC, and are shown as follows:

- Completed assignment statements are identified using [*italicized text within brackets*].
- Completed selection statements are identified using [underlined text within brackets]. In keeping with these conventions, in the event an assignment is within a selection, it will be depicted as *italicized, underlined* text.
- Refinements are identified using **bold text**. Any text removed is stricken (Example: ~~TSF Data~~) and should be considered as a refinement. In keeping with these conventions, in the event a refinement is within an assignment, it will be depicted as ***bold italicized*** text, and when a refinement is within a selection, it will be depicted in **bold underlined** text.
- Extended Functional and Assurance Requirements are identified using "_EXT" at the end of the short name.
- Iterations are identified by appending a number in parentheses following the component title. For example, FAU_GEN.1(1) Audit Data Generation would be the first iteration and FAU_GEN.1(2) Audit Data Generation would be the second iteration.

6.2 Security Functional Requirements

This section specifies the SFRs for the TOE. This section organizes the SFRs by CC class. Table 11 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement.

Table 11 TOE Security Functional Requirements

Name	Description	S	A	R	I
FAU_GEN.1	Audit data generation	✓	✓		
FAU_GEN.2	User identity association				
FAU_STG_EXT.1	External Audit Trail Storage	✓			
FCS_CKM.1	Cryptographic key generation (for asymmetric keys)	✓		✓	
FCS_CKM_EXT.4	Cryptographic Key Zeroization				
FCS_COP.1(1)	Cryptographic operation (for data encryption/decryption)	✓	✓	✓	✓
FCS_COP.1(2)	Cryptographic operation (for cryptographic signature)	✓		✓	✓
FCS_COP.1(3)	Cryptographic operation (for cryptographic hashing)	✓	✓	✓	✓
FCS_COP.1(4)	Cryptographic operation (for keyed-hash message authentication)	✓	✓	✓	✓

Name	Description	S	A	R	I
FCS_HTTPS_EXT.1	Explicit: HTTPS				
FCS_RBG_EXT.1	Extended: Cryptographic Operation (Random Bit Generation)	✓			
FCS_TLS_EXT.1	Explicit: TLS	✓			
FDP_RIP.2	Full Residual Information Protection	✓			
FIA_PMG_EXT.1	Password Management				
FIA_UAU.7	Protected Authentication Feedback		✓		
FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism	✓	✓		
FIA_UIA_EXT.1	User Identification and Authentication	✓	✓		
FMT_MTD.1(1)	Management of TSF data (for general TSF data)	✓	✓		✓
FMT_MTD.1(2)	Management of TSF data (for cryptographic information)	✓	✓		✓
FMT_MTD.1(3)	Management of TSF data (for administrator accounts)	✓	✓		✓
FMT_SMF.1	Specification of management functions	✓	✓		
FMT_SMR.2	Restrictions on Security Roles	✓	✓		
FPT_APW_EXT.1	Extended: Protection of Administrator Passwords				
FPT_ITT.1	Basic Internal TSF Data Transfer Protection	✓	✓	✓	
FPT_SKP_EXT.1	Extended: Protection of TSF Data (for reading of all symmetric keys)				
FPT_STM.1	Reliable Time Stamps				
FPT_TST_EXT.1	TSF testing				
FPT_TUD_EXT.1	Extended: Trusted Update	✓			
FTA_SSL.3	TSF-initiated Termination		✓	✓	
FTA_SSL.4	User-initiated Termination				
FTA_SSL_EXT.1	TSF-initiated session locking	✓			
FTA_TAB.1	Default TOE access banners			✓	
FTP_ITC.1	Inter-TSF Trust Channel		✓	✓	✓
FTP_TRP.1	Trusted Path	✓	✓	✓	✓

Note: S=Selection; A=Assignment; R=Refinement; I=Iteration

6.2.1 Class FAU: Security Audit

FAU_GEN.1 Audit Data Generation
Hierarchical to: No other components.
Dependencies: FPT_STM.1 Reliable time stamps
FAU_GEN.1.1

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

a) Start-up and shutdown of the audit functions;
b) All auditable events, for the [not specified] level of audit; and
c) [*All administrative actions*]
d) [*Specifically defined auditable events listed in Table 12*].

Table 12 Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM_EXT.4	None.	None.
FCS_COP.1(1)	None.	None.
FCS_COP.1(2)	None.	None.
FCS_COP.1(3)	None.	None.
FCS_COP.1(4)	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_TLS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session. Establishment/Termination of a HTTPS session.	Reason for Failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FDP_RIP.2	None.	None.
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FMT_MTD.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_ITT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_STM.1	Changes to the time.	The old and new values for the time Origin of the attempt (e.g. IP address)
FPT_TUD_EXT.1	Initiation of update.	No additional information.
FPT_TST_EXT.1	None.	None.
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.
FTA_SSL.4	The termination of an interactive session.	No additional information.
FTA_TAB.1	None.	None.
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.

FAU_GEN.1.2

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column three of Table 12*].

FAU_GEN.2 User identity association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

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.

FAU_STG_EXT.1 External Audit Trail 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 implementing the [TLS/HTTPS] protocol.

6.2.2 Class FCS: Cryptographic Support

FCS_CKM.1 Cryptographic key generation
Hierarchical to: No other components.
Dependencies: FCS_COP.1 Cryptographic operation
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1

Refinement: The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with: [NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptical curve-based key establishment schemes, and implementing "NIST curves" P-256, P-384, and [P-521] (as defined in FIPS PUB 186-3, "Digital Signature Standard").]and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

FCS_CKM_EXT.4 Cryptographic key destruction
Hierarchical to: FCS_CKM.4
Dependencies: FCS_CKM.1 Cryptographic key generation

FCS_CKM_EXT.4.1

The TSF shall zeroize all plaintext secret and private cryptographic keys and CSP[23]s when no longer required.

FCS_COP.1(1) Cryptographic operation (for data encryption/decryption)
Hierarchical to: No other components.
Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1(1).1

The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [***AES operating in [ECB[24], CBC[25], CFB[26](128), OFB[27](128), and CTR[28](128-, 192-, 256-bit key sizes) modes]***] and cryptographic key sizes 128-bits, 256-bits, and [**192 bits**] that meet the following:

- **FIPS PUB 197, "Advanced Encryption Standard (AES)"**
- **[NIST SP 800-38A]**

FCS_COP.1(2) Cryptographic operation (for cryptographic signature)
Hierarchical to: No other components.
Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1(2).1

Refinement: The TSF shall perform **cryptographic signature services** in accordance with a [
(1) Digital Signature Algorithm (DSA) with a key size (modulus) of 2048 bits or greater, or
(2) Elliptical Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater]
that meets the following:

- FIPS PUB 186-3, "Digital Signature Standard"
- The TSF shall implement "NIST curves" P-256, P-384 and [P-521] as defined in FIPS PUB 186-3, "Digital Signature Standard").

[23] CSP – Critical Security Parameters
[24] ECB – Electronic Codebook
[25] CBC – Cipher Block Chaining
[26] CFB – Cipher Feedback
[27] OFB – Output Feedback
[28] CTR – Counter mode

FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)
Hierarchical to: No other components.
Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1(3).1

The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [**SHA-1, SHA-256, SHA-384, SHA-512**] **and message digest sizes** [**160, 256, 384, 512**] **bits** that meet the following: FIPS Pub 180-3, "Secure Hash Standard."

FCS_COP.1(4) Cryptographic operation (for keyed-hash message authentication)
Hierarchical to: No other components.
Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1(4).1

Refinement: The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC-[**SHA-1, SHA-224, SHA-256, SHA-384, SHA-512**], **and message digest sizes** [**160, 224, 256, 384, 512**] **bits** that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."

FCS_HTTPS_EXT.1 Explicit:HTTPS
Hierarchical to: No other components.
Dependencies: FCS_TLS_EXT.1

FCS_HTTPS_EXT.1.1

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

FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)
Hierarchical to: No other components.
Dependencies: None.

FCS_RBG_EXT.1.1

The TSF shall perform all random bit generation (RBG) services in accordance with [NIST Special Publication 800-90 using [HMAC DRBG (any)];] seeded by an entropy source that accumulated entropy from [a software-based noise source; a TSF-hardware-based noise source].

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded with a minimum of [256 bits] of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

FCS_TLS_EXT.1 Explicit:TLS
Hierarchical to: No other components.
Dependencies: FCS_COP.1(1) Cryptographic operation (for data encryption/decryption).

FCS_TLS_EXT.1.1

The TSF shall implement one or more of the following protocols [TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:
Mandatory Ciphersuites:
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
Optional Ciphersuites:
[*None*].

6.2.3 Class FDP: User Data Protection

FDP_RIP.2 Full Residual Information Protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_RIP.2.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon the [deallocation of the resource from] all objects.

6.2.4 Class FIA: Identification and Authentication

FIA_PMG_EXT.1 Password Management
Hierarchical to: No other components.
Dependencies: No dependencies.
FIA_PMG_EXT.1.1
The TSF shall provide the following password management capabilities for administrative passwords:
1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [“`”, “~”, “-“, “+”, “=”, “[“, “]”, “\”, “|”, “;”, “:”, “<”, “>”, “.”, “/”, “?”, *comma, quotation mark, underscore, tab, space*]];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism
Hierarchical to: No other components.
Dependencies: No dependencies.
FIA_UAU_EXT.2.1
The TSF shall provide a local password-based authentication mechanism, [*none*] to perform administrative user authentication.

FIA_UAU.7 Protected Authentication Feedback
Hierarchical to: No other components.
Dependencies: FIA_UAU.1
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.

FIA_UIA_EXT.1 User Identification and Authentication
Hierarchical to: FIA_UID.1 Timing of identification
FIA_UAU.1 Timing of Authentication
Dependencies: No dependencies.
FIA_UIA_EXT.1.1
The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:
- Display the warning banner in accordance with FTA_TAB.1;
- [[*no other actions.*]]

FIA_UIA_EXT.1.2
The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

6.2.5 Class FMT: Security Management

FMT_MTD.1(1) Management of TSF Data (for general TSF data)
Hierarchical to: No other components.
Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles
FMT_MTD.1(1).1
The TSF shall restrict the ability to *manage* the *TSF data* to the *Security Administrators*.

FMT_MTD.1(2) Management of TSF Data (for cryptographic information)
Hierarchical to: No other components.
Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles
FMT_MTD.1(2).1
The TSF shall restrict the ability to [*create, initialize, view, change_default, modify, delete, clear, and append*] the [*cryptographic information*] to the [*Administrators*].

FMT_MTD.1(3) Management of TSF Data (for administrator accounts)
Hierarchical to: No other components.
Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles
FMT_MTD.1(3).1
The TSF shall restrict the ability to [*create, view, modify, delete*] the [*Administrator Accounts*] to the [*Administrators*].

FMT_SMF.1 Specification of Management Functions
Hierarchical to: No other components.
Dependencies: FIA_UIA_EXT.1 User Identification and Authentication
FCS_COP.1(2) Cryptographic operation (for cryptographic signature)
FPT_TUD_EXT.1 Extended: Trusted Update
FMT_SMF.1.1
The TSF shall be capable of performing the following management functions:[
- Ability to Administer the TOE locally and remotely;
- Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;
- [Ability to configure the cryptographic functionality].

FMT_SMR.2 Security roles
Hierarchical to: No other components.
Dependencies: FIA_UID.1 Timing of identification
FMT_SMR.2.1
The TSF shall maintain the roles:
- **Authorized Administrator**[29]

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

are satisfied.

[29] Authorized roles include sysadmin and Administrators.

6.2.6 Class FPT: Protection of the TSF

FPT_ITT.1 Basic Internal TSF Data Transfer Protection
Hierarchical to: No other components.
Dependencies: FCS_TLS_EXT.1 Explicit TLS
FCS_HTTPS_EXT.1 Explicit HTTPS
FPT_ITT.1.1

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

FPT_SKP_EXT.1 Extended: Protection of TSF Data (for reading of all symmetric keys)
Hierarchical to: No other components.
Dependencies: No dependencies.
FPT_SKP_EXT.1.1
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

FPT_APW_EXT.1 Extended: Protection of Administrator Passwords
Hierarchical to: No other components.
Dependencies: No dependencies.
FPT_APW_EXT.1.1
The TSF shall store passwords in non-plaintext form.
FPT_APW_EXT.1.2
The TSF shall prevent the reading of plaintext passwords.

FPT_STM.1 Reliable Time Stamps
Hierarchical to: No other components.
Dependencies: No dependencies.
FPT_STM.1.1

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

FPT_TUD_EXT.1 Extended: Trusted Update
Hierarchical to: No other components.
Dependencies: FCS_COP.1(2) Cryptographic operation (for cryptographic signature)
FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)
FPT_TUD_EXT.1.1
The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.
FPT_TUD_EXT.1.2
The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.
FPT_TUD_EXT.1.3
The TSF shall provide a means to verify firmware/software updates to the TOE using a [digital signature mechanism] prior to installing those updates.

FPT_TST_EXT.1 TSF testing
Hierarchical to: No other components.
Dependencies: No dependencies.
FPT_TST_EXT.1.1
The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

6.2.7 Class FTA: TOE Access

FTA_SSL_EXT.1 TSF-initiated session locking
Hierarchical to: No other components.
Dependencies: FIA_UAU.1 Timing of authentication
FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [terminate the session] after a Security Administrator-specified time period of inactivity.

FTA_SSL.3 TSF-initiated termination
Hierarchical to: No other components.
Dependencies: No dependencies.
FTA_SSL.3.1

Refinement: The TSF shall terminate **a remote** interactive session after a [*Administrator-configurable time interval of user inactivity*].

FTA_SSL.4 User-initiated Termination
Hierarchical to: No other components.
Dependencies: No dependencies.
FTA_SSL.4.1

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

FTA_TAB.1 Default TOE access banners
Hierarchical to: No other components.
Dependencies: No dependencies.
FTA_TAB.1.1

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

6.2.8 Class FTP: Trusted Path/Channels

FTP_ITC.1 Inter-TSF trusted channel
Hierarchical to: No other components.
Dependencies: FCS_TLS_EXT.1 Explicit TLS

FTP_ITC.1.1

Refinement: The TSF shall **use [**TLS**] to** provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [[*no other capabilities*]]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data**.

FTP_ITC.1.2

Refinement: The TSF shall permit *the TSF,* **or the authorized IT entities** to initiate communication via the trusted channel.

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for [*syslog*].

FTP_TRP.1 Trusted Path
Hierarchical to: No other components.
Dependencies: FCS_HTTPS_EXT.1 Explicit HTTPS
FCS_TLS_EXT.1 Explicit TLS

FTP_TRP.1.1

Refinement: The TSF shall **use [TLS/HTTPS]** provide **a trusted** communication path between itself and **remote administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **disclosure and detection of modification of the communicated data**.

FTP_TRP.1.2

Refinement: The TSF shall permit **remote administrators** to initiate communication via the trusted path.

FTP_TRP.1.3

The TSF shall require the use of the trusted path for [*initial administrator authentication and all remote administrator actions*].

6.3 Security Assurance Requirements

This section defines the assurance requirements for the TOE. Assurance requirements are taken from NDPP v1.1 Section 4.3. Table 13 below summarizes the requirements.

Table 13 NDPP v1.1 Assurance Requirements

Assurance Requirements	
Class ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.1 Security objectives for the operational environment
	ASE_REQ.1 Stated security requirements
	ASE_TSS.1 TOE summary specification
Class ALC : Life Cycle Support	ALC_CMC.1 Labeling of the TOE
	ALC_CMS.1 TOE CM[30] Coverage
Class ADV: Development	ADV_FSP.1 Basic functional specification
Class AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
Class ATE: Tests	ATE_IND.1 Independent testing – conformance
Class AVA: Vulnerability assessment	AVA_VAN.1 Vulnerability survey

[30] CM – Configuration Management

7 TOE Summary Specification

This section presents information to detail how the TOE meets the functional requirements described in previous sections of this ST.

7.1 TOE Security Functions

Each of the security requirements and the associated descriptions correspond to the security functions. Hence, each function is described by how it specifically satisfies each of its related requirements. This serves to both describe the security functions and rationalize that the security functions satisfy the necessary requirements.

Table 14 Mapping of TOE Security Functions to Security Functional Requirements

TOE Security Function	SFR ID	Description
Security Audit	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User identity association
	FAU_STG_EXT.1	External Audit Trail Storage
Cryptographic Support	FCS_CKM.1	Cryptographic key generation (for asymmetric keys)
	FCS_CKM_EXT.4	Cryptographic Key Zeroization
	FCS_COP.1(1)	Cryptographic operation (for data encryption/decryption)
	FCS_COP.1(2)	Cryptographic operation (for cryptographic signature)
	FCS_COP.1(3)	Cryptographic operation (for cryptographic hashing)
	FCS_COP.1(4)	Cryptographic operation (for keyed-hash message authentication)
	FCS_HTTPS_EXT.1	Explicit: HTTPS
	FCS_RBG_EXT.1	Extended: Cryptographic Operation (Random Bit Generation)
	FCS_TLS_EXT.1	Explicit: TLS
User Data Protection	FDP_RIP.2	Full Residual Information Protection
Identification and Authentication	FIA_PMG_EXT.1	Password Management
	FIA_UAU.7	Protected Authentication Feedback
	FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism
	FIA_UIA_EXT.1	User Identification and Authentication

TOE Security Function	SFR ID	Description
Security Management	FMT_MTD.1(1)	Management of TSF data (for general TSF data)
	FMT_MTD.1(2)	Management of TSF data (for cryptographic information)
	FMT_MTD.1(3)	Management of TSF data (for administrator accounts)
	FMT_SMF.1	Specification of management functions
	FMT_SMR.2	Restrictions on Security Roles
Protection of the TSF	FPT_APW_EXT.1	Extended: Protection of Administrator Passwords
	FPT_ITT.1	Basic Internal TSF Data Transfer Protection
	FPT_SKP_EXT.1	Extended: Protection of TSF Data (for reading of all symmetric keys)
	FPT_STM.1	Reliable Time Stamps
	FPT_TST_EXT.1	TSF testing
	FPT_TUD_EXT.1	Extended: Trusted Update
TOE Access	FTA_SSL.3	TSF-initiated Termination
	FTA_SSL.4	User-initiated Termination
	FTA_SSL_EXT.1	TSF-initiated session locking
	FTA_TAB.1	Default TOE access banners
Trusted path/channels	FTP_ITC.1	Inter-TSF Trust Channel
	FTP_TRP.1	Trusted Path

7.1.1 Security Audit

The Security Audit function provides the TOE with the functionality of generating audit records. As administrators manage and configure the TOE, their activities are tracked and recorded as audit records and are stored in the file system. The resulting audit records can be examined to determine which security relevant activities took place and who (i.e., which user) is responsible for those activities.

The TOE provides auditing of administrator actions that occur within the SMC. For audit events that result from actions of identified users, the TOE associates the action with the user who took the action in the logs. The SMC provides an authorized administrator access to view the audit logs created as a result of administrator actions through the SMC and via the reporting features. In the SMC, the Audit Log page details the audit events. Only authorized administrators with the appropriate role and permissions can review the security audit logs.

Audit Log entries contain the following fields:

- Timestamp – Date and Time that the event occurred,

- Category – The category with which the event is associated,
- Event – A brief description of the audited event,
- Message Text – Message text that describes the event or provides more information,
- User – Login name of the user associated with the audited action,
- User Location – IP address of the device the user used to perform the action,
- Process Name – The component that issued the log message,
- Success – Indicates whether the action was completed or not.

Not all of these fields may be present for every auditable event. For example, a log of Central Processing Unit (CPU) reaching capacity would not have a User associated because the User does not directly affect CPU usage.

For each of the communication protocols used, TLS and HTTPS sessions that fail due to user authentication failures are auditable. No other protocol failures are audited.

The TOE transfers all log data as it is generated to a remote syslog server for external storage therefore only one megabyte of local storage is allocated per appliance. Syslog data is protected via an encrypted TLS tunnel. All encryption is provided by CAVP-validated algorithms. If the log file reaches one megabyte of storage the log file is rotated. Upon rotation, all existing log data is exported and the log file is cleared. New log records can now be written into the log file.

TOE Security Functional Requirements Satisfied: FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1.

7.1.2 Cryptographic Support

Cryptographic operations on the TOE are provided by a FIPS 140-2 validated cryptographic module[31] (for TLS, HTTPS, and key generation). The TOE uses TLS/HTTPS to protect communications. HTTPS is used to encrypt management connections via the SMC and communications between physically separate TOE components. The TOE can use AES cryptographic algorithm to encrypt and decrypt data. The TOE also provides SHA, HMAC-SHA, and SHS to support TOE cryptographic functionality.

The TOE's cryptographic module is capable of generating cryptographic keys that provide at least 112 bits of symmetric key strength, in accordance with FIPS standards. Keys are generated via the use of an HMAC DRBG to provide random keying material in accordance with NIST Special Publication 800-90A. When the TOE is finished using a cryptographic key, the key is zeroized.

The TOE can use AES 128 and 256-bit when processing HTTPS/TLS requests depending on the capabilities of the client. When establishing a session, the client and server use the standard TLS handshake protocol, which involves exchanging the server's certificate and then the client returning an encrypted pre-master secret. The client and server then use the pre-master-secret to generate keys known only to the client and server. These keys are used to encrypt all future messages between the client and server. HTTPS/TLS is used for management sessions via the web interface, communications via the Web API, and protecting communications between physically separate TOE components. TLS is used to protect communications with a remote syslog server. The TOE uses the following ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA

The TOE uses a DSA key in a digital certificate to perform key exchange with clients connecting via HTTPS. This key is loaded by default, but can be replaced with another certificate manually input by an administrator, or another certificate can be generated by the TOE. All symmetric keys are AES keys,

[31] Certificate #1291. The FIPS validation is vendor affirmed and has been ported according to FIPS IG G.5.

including the keys for HTTPS, TLS, and inter-TOE communications. No other keys or key-generating CSPs are used by the TOE. Certificate keys are only zeroized when the certificate expires or when the certificates are replaced. AES keys are zeroized after the session they are associated with ends. Zeroization is done for all keys by overwriting all key data with zeros. Only TLS certificate keys are stored persistently, and these are also overwritten with zeros upon zeroization.

TOE Security Functional Requirements Satisfied: FCS_CKM.1, FCS_CKM_EXT.4, FCS_COP.1, FCS_HTTPS_EXT.1, FCS_RBG_EXT.1, FTC_TLS_EXT.1.

7.1.3 User Data Protection

The TOE clears the memory space used for storing network data by overwriting the memory space with zeros after the TOE finishes using that memory space. The TOE ensures that no residual data remains prior to allocation of memory, ensuring that any attempt to reconstruct the content of the memory buffers after reallocation will result in the reconstruction of the zeros, not packet data. Buffers are cleared by overwriting all existing data with zeros before allocating the buffer space to a new connection.

TOE Security Functional Requirements Satisfied: FDP_RIP.2.

7.1.4 Identification and Authentication

The CLI is utilized in accessing this function. Users can view the login banner prior to authenticating to the TOE. The TOE must perform successful identification and authentication of the TOE administrative user before the TSF grants the user access to other TOE security functions on the CLI or web interface. Administrator user authentication is enforced through the use of a password. Passwords must meet the following criteria:

- composed of upper- and lower-case letters, numbers, and special characters,
- minimum password length of 15 characters,

While authenticating via the web interface, the TOE obscures the User's password so that none of the characters are visible while being typed. While authenticating via the CLI, the TOE does not provide any visual feedback for the User's password. When a User's password expires, the User is required to input a new password after entering the expired password.

Authentication via both methods (CLI and web interface) requires the use of a username and password combination. The CLI only accepts credentials via a serial connection and the web interface only accepts credentials via HTTPS. A login is considered successful if the username and the SHA-512 hash of the User's password match the stored username and password hash stored on the TOE.

TOE Security Functional Requirements Satisfied: FIA_PMG_EXT.1, FIA_UIA_EXT.1, FIA_UAU_EXT.2, FIA_UAU.7.

7.1.5 Security Management

Security management specifies how the TOE manages several aspects of the TSF including TSF data and security functions. TSF data includes configuration data of the TSF and audit data, cryptographic functionality and information, hosts, dashboards and analytics, and administrator accounts. The TOE provides authorized administrators with a web GUI to easily manage the security functions and TSF data of the TOE. The GUI can be used to configure the cryptographic functionality available on the TOE, update the TOE, and verify the updates via digital signatures.

The TOE defines two Authorized Administrator roles: Administrator and sysadmin. There is also a User role that does not have access to management functionality. Administrators have custom permissions that define their level of access to management functionality and data stored by the TOE. The role Administrator performs all functions listed for a Security Administrator. The sysadmin role has access only to the CLI. The CLI is accessed locally. The Administrator role has remote access to the TOE only.

Unauthenticated users only have access to read the displayed warning banner before authenticating successfully with the TOE. While the TOE access banner is displayed to all users before authentication, it is read-only and cannot be modified by an unauthenticated user (and in fact is not modifiable from the login screen at all).

TOE Security Functional Requirements Satisfied: FMT_MTD.1(1), FMT_MTD.1(2), FMT_MTD.1(3), FMT_SMF.1, FMT_SMR.2.

7.1.6 Protection of the TSF

The TOE provides HTTPS/TLS and TLS to protect TSF data from disclosure and detecting modification while in transit between different parts of the TOE.

The TOE does not allow any User or Administrator to read plaintext passwords stored on the TOE, since all passwords are stored in hashed form using SHA-512. The TOE also prevents pre-shared, symmetric, and private keys from being read by storing keys in internally-allocated data structures. This means that key data, which is stored in volatile memory in plaintext, can only be output via the cryptographic library's API, and no User-accessible interfaces can be used to read keys. The Operating System (OS) and Java Runtime Environment (JRE) safeguard memory and process space from unauthorized access. Direct access to memory can only occur through the CLI, which can only be accessed by an authorized administrator.

The TOE generates its own time stamps that originate from a system hardware clock. The timestamp is used by the audit logs to record an accurate time for each auditable event. The time can be changed through the Web UI. An authorized administrator can go to the **Configuration > System Time and NTP** page to modify the time. Use of an NTP server is not part of the evaluated configuration. Once a time change is made the system must be restarted.

Administrators can find the current version of TOE software by going to the home page of the Web UI. The TOE also provides a feature to update the TOE. Update files can be verified with a digital signature. The TOE is shipped with two certificates stored internally that are used to verify the signature on update files (no live key exchange is used) thereby guaranteeing that the updates are both valid and trusted. At power up, the TOE runs a suite of self-tests that check for the correct operation of the cryptographic functionality provided by the cryptographic module. All hardware models run these tests on startup. A description of each self-test is given in Table 1 below.

Table 15 Self-Test Descriptions

Self-Test	Description
AES KAT	The AES KAT encrypts a known plaintext with known keys. It then compares the resultant ciphertext with the expected ciphertext hard-coded in the TOE. If the two values differ, then the KAT fails. If the two values agree, the AES KAT then decrypts the ciphertext with the known keys and compares the decrypted text with the known plaintext. If they differ, then the test fails. If they are the same, then the test passes.
SHA-1 KAT	The KAT takes a specific value and hashes it. This digest value is then compared to the known value. If the values differ, the test fails. If they are the same, the test passes.
SHA-224 KAT	The KAT takes a specific value and hashes it. This digest value is then compared to the known value. If the values differ, the test fails. If they are the same, the test passes.
SHA-256 KAT	The KAT takes a specific value and hashes it. This digest value is then compared to the known value. If the values differ, the test fails. If they are the same, the test passes.
SHA-384 KAT	The KAT takes a specific value and hashes it. This digest value is then compared to the known value. If the values differ, the test fails. If they are the same, the test passes.
SHA-512 KAT	The KAT takes a specific value and hashes it. This digest value is then compared to the known value. If the values differ, the test fails. If they are the same, the test passes.
HMAC SHA-1 KAT	The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes.
HMAC SHA-224 KAT	The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes.
HMAC SHA-256 KAT	The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes.
HMAC SHA-384 KAT	The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes.
HMAC SHA-512 KAT	The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes.
HMAC DRBG Self-Test	A known seed value is used to initialize the DRBG. A block of random data is then generated and compared to a pre-generated value. If these values are the same, the test is passed. Otherwise, the test is failed.
EC DRBG Self-Test	A known seed value is used to initialize the DRBG. A block of random data is then generated and compared to a pre-generated value. If these values are the same, the test passes. Otherwise, the test fails.

If a cryptographic self-test fails the FIPS module enters an error state which is reported to the TOE. The TOE's FIPS state will move to disabled. An Administrator must re-enable FIPS mode to clear the error.

The FIPS module runs self-tests on all algorithms found within the module. The TOE is configured to use only the HMAC DRBG, but all DRBGs are tested at start-up.

TOE Security Functional Requirements Satisfied: FPT_ITT.1, FPT_SKP_EXT.1, FPT_APW_EXT.1, FPT_STM.1, FPT_TUD_EXT.1, FPT_TST_EXT.1.

7.1.7 TOE Access

The TOE terminates local and remote management sessions after an Administrator configurable time period of inactivity. Administrators may also terminate their sessions voluntarily. Users must log in again to regain access to TOE management capabilities. At the login screen Users are shown an advisory notice and consent warning message regarding unauthorized use of the TOE. The message is shown to users of both the web interface and the CLI.

TOE Security Functional Requirements Satisfied: FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4, FTA_TAB.1.

7.1.8 Trusted Path/Channels

The TOE provides a trusted path between the TOE management interfaces and remote TOE administrators. These interfaces are the Web Services API, Client Interface and Web Admin GUI. These interfaces are protected via HTTPS. These protocols and the cryptography they implement provide adequate defense against unauthorized disclosure and detection of modification of data being communicated. Additionally, the TOE protects syslog traffic by encrypting it with a secure TLS tunnel. This tunnel prevents unauthorized disclosure and detection of modification for all audit data sent to the remote syslog server. The TOE does not communicate with any other servers or network devices in the evaluated configuration.

TOE Security Functional Requirements Satisfied: FTP_ITC.1, FTP_TRP.1.

8 Rationale

8.1 Conformance Claims Rationale

This Security Target conforms to Part 2 extended and Part 3 conformant of the Common Criteria Standard for Information Technology Security Evaluations, Version 3.1 Revision 4. This ST conforms to the NDPP v1.1.

8.1.1 Variance Between the PP and this ST

In some instances changes were made in this ST from the NDPP. All of these changes are documented below with a rationale for the change.

- An Application Note in the NDPP states that the word "manage" in FMT_MTD.1 is the default requirement for management of TSF data. Other iterations are possible. Iterations were added for specific management functions.

8.1.2 Security Assurance Requirements Rationale

This ST maintains exact conformance to NDPP v1.1, including the assurance requirements listed in section 4.3 of NDPP.

8.1.3 Dependency Rationale

This ST does satisfy all the requirement dependencies of the Common Criteria. Table 16 lists each requirement to which the TOE claims conformance with a dependency and indicates whether the dependent requirement was included. As the table indicates, all dependencies have been met.

Table 16 Functional Requirements Dependencies

SFR ID	Dependencies	Dependency Met	Rationale
FAU_GEN.1	FPT_STM.1	✓	
FAU_GEN.2	FIA_UID.1	✓	Although FIA_UID.1 is not included, FIA_UIA_EXT.1 provides coverage for user identification and authentication which supersedes FIA_UID.1.
	FAU_GEN.1	✓	
FAU_STG_EXT.1	FAU_GEN.1	✓	
FCS_CKM.1	FCS_COP.1(1)	✓	
	FCS_COP.1(2)	✓	
	FCS_COP.1(3)	✓	
	FCS_COP.1(4)	✓	
	FCS_CKM.4	✓	Although FCS_CKM.4 is not in the ST, FCS_CKM_EXT.4 provides coverage.

SFR ID	Dependencies	Dependency Met	Rationale
FCS_CKM_EXT.4	FCS_CKM.1	✓	
FCS_COP.1(1)	FCS_CKM.4	✓	Although FCS_CKM.4 is not in the ST, FCS_CKM_EXT.4 provides coverage.
	FCS_CKM.1	✓	
FCS_COP.1(2)	FCS_CKM.4	✓	Although FCS_CKM.4 is not in the ST, FCS_CKM_EXT.4 provides coverage.
	FCS_CKM.1	✓	
FCS_COP.1(3)	FCS_CKM.1	✓	
	FCS_CKM.4	✓	Although FCS_CKM.4 is not in the ST, FCS_CKM_EXT.4 provides coverage.
FCS_COP.1(4)	FCS_CKM.4	✓	Although FCS_CKM.4 is not in the ST, FCS_CKM_EXT.4 provides coverage.
	FCS_CKM.1	✓	
FCS_HTTPS_EXT.1	FCS_TLS_EXT.1	✓	
FCS_RBG_EXT.1	No dependencies	✓	
FCS_TLS_EXT.1	FCS_COP.1(1)	✓	
FDP_RIP.2	No dependencies	✓	
FIA_PMG_EXT.1	No dependencies	✓	
FIA_UAU.7	FIA_UAU.1	✓	Although FIA_UAU.1 is not included, FIA_UIA_EXT.1 provides coverage for user identification and authentication which supersedes FIA_UAU.1.
FIA_UAU_EXT.2	No dependencies	✓	
FIA_UIA_EXT.1	No dependencies	✓	
FMT_MTD.1(1)	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MTD.1(2)	FMT_SMR.1	✓	
	FMT_SMF.1	✓	
FMT_MTD.1(3)	FMT_SMF.1	✓	

SFR ID	Dependencies	Dependency Met	Rationale
	FMT_SMR.1	✓	
FMT_SMF.1	FPT_TUD_EXT.1	✓	
	FIA_UIA_EXT.1	✓	
	FCS_COP.1(2)	✓	
FMT_SMR.2	FIA_UID.1	✓	Although FIA_UID.1 is not included, FIA_UIA_EXT.1 provides coverage for user identification and authentication which supersedes FIA_UID.1.
FPT_APW_EXT.1	No dependencies	✓	
FPT_ITT.1	FCS_TLS_EXT.1	✓	
	FCS_HTTPS_EXT.1	✓	
FPT_SKP_EXT.1	No dependencies	✓	
FPT_STM.1	No dependencies	✓	
FPT_TST_EXT.1	No dependencies	✓	
FTA_SSL.3	No dependencies	✓	
FTA_SSL.4	No dependencies	✓	
FTA_SSL_EXT.1	FIA_UAU.1	✓	Although FIA_UAU.1 is not included, FIA_UIA_EXT.1 provides coverage for user identification and authentication which supersedes FIA_UAU.1.
FTA_TAB.1	No dependencies	✓	
FTP_ITC.1	FCS_TLS_EXT.1	✓	
FTP_TRP.1	FCS_SSH_EXT.1	✓	
	FCS_TLS_EXT.1	✓	
	FCS_HTTPS_EXT.1	✓	

9 Acronyms and Terms

This section describes the acronyms and terms.

9.1 Terminology

Table 17 Terms

Name	Definition
Authorized Administrator	A user with administrator TOE access that has been successfully identified and authenticated by the TOE.
Domain parameters	DSA requires that the private/public key pairs used for digital signature generation and verification be generated with respect to a particular set of domain parameters. These domain parameters may be common to a group of users and may be public. A user of a set of domain parameters (i.e., both the signatory and the verifier) shall have assurance of their validity prior to using them. Although domain parameters may be public information, they shall be managed so that the correct correspondence between a given key pair and its set of domain parameters is maintained for all parties that use the key pair. A set of domain parameters may remain fixed for an extended time period. The domain parameters for DSA are the integers p, q, and g, and optionally, the domain_parameter_seed and counter that were used to generate p and q (i.e., the full set of domain parameters is (p, q, g {, domain_parameter_seed, counter})).
Hardware-based noise source	A hardware random number generator is an apparatus that generates random numbers from a physical process. Such devices are often based on microscopic phenomena that generate a low-level, statistically random "noise" signal, such as thermal noise or the photoelectric effect or other quantum phenomena. These processes are, in theory, completely unpredictable, and the theory's assertions of unpredictability are subject to experimental test. A hardware random number generator typically consists of a transducer to convert some aspect of the physical phenomena to an electrical signal, an amplifier and other electronic circuitry to increase the amplitude of the random fluctuations to a macroscopic level, and some type of analog to digital converter to convert the output into a digital number, often a simple binary digit 0 or 1. By repeatedly sampling the randomly varying signal, a series of random numbers is obtained.
Target network	The domain of network and managed devices to be analyzed by the TOE.

9.2 Acronyms

Table 18 Acronyms

Acronym	Definition
AEAD	Authenticated Encryption with Additional Authenticated Data
AES	Advanced Encryption Standard
ANSI	American National Standards Institute

Acronym	Definition
API	Application Programming Interface
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CEM	Common Evaluation Methodology
CFB	Cipher Feedback
CLI	Command Line Interface
CM	Configuration Management
CPU	Central Processing Unit
CSP	Critical Security Parameters
CTR	Counter Mode
DES	Data Encryption Standard
DPI	Deep Packet Inspection
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
EC	Elliptical Curve
ECB	Electronic Code Book
ECDRBG	Elliptical Curve Deterministic Random Bit Generator
ECDSA	Elliptical Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standard
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
ID	Identifier
IP	Internet Protocol
IPFIX	Internet Protocol Flow Information Export
IPSEC	Internet Protocol Security
IT	Information Technology
JRE	Java Runtime Environment
MAC	Message Authentication Code
MPLS	Multiprotocol Label Switching

Acronym	Definition
NDPP	Network Device Protection Profile
NIST	National Institute of Standards and Technology
OFB	Output Feedback
OS	Operating System
OSP	Organizational Security Policy
PP	Protection Profile
PUB	Publication
RBG	Random Bit Generator
RFC	Request for Comment
RSA	Rivest, Shamir, and Adelman
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SMC	StealthWatch Management Console
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SP	Security Policy
SPAN	Switch Port Analyzer
SSH	Secure Shell
ST	Security Target
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
UDP	User Datagram Protocol
VE	Virtual Edition

Prepared by:
Corsec Security, Inc.

13135 Lee Jackson Memorial Highway
Suite 220
Fairfax, VA 22033
United States of America

Phone: +1 (703) 267-6050
Email: info@corsec.com
http://www.corsec.com