Blue Coat® Systems, Inc.
SSL Visibility Appliance

Models: SV1800-C, SV1800-F, SV2800, SV3800
Hardware Versions: 090-03061, 090-03062, 090-03063, and 090-03064
Software Version: 3.8.4FC
Evaluation Assurance Level: EAL 3+ Augmented with ALC_FLR.3

Blue Coat® Systems, Inc. SSL Visibility Appliance EAL 3+ Augmented with ALC_FLR.3 Security Target

Document Version: 0.26

COPYRIGHT NOTICE

BLUE COAT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. BLUE COAT PRODUCTS, TECHNICAL SERVICES, AND ANY OTHER TECHNICAL DATA REFERENCED IN THIS DOCUMENT ARE SUBJECT TO U.S. EXPORT CONTROL AND SANCTIONS LAWS, REGULATIONS AND REQUIREMENTS, AND MAY BE SUBJECT TO EXPORT OR IMPORT REGULATIONS IN OTHER COUNTRIES. YOU AGREE TO COMPLY STRICTLY WITH THESE LAWS, REGULATIONS AND REQUIREMENTS, AND ACKNOWLEDGE THAT YOU HAVE THE RESPONSIBILITY TO OBTAIN ANY LICENSES, PERMITS OR OTHER APPROVALS THAT MAY BE REQUIRED IN ORDER TO EXPORT, RE-EXPORT, TRANSFER IN COUNTRY OR IMPORT AFTER DELIVERY TO YOU.

Americas: Rest of the World:

Blue Coat Systems, Inc.
420 N. Mary Ave
Sunnyvale, CA 94085

Blue Coat Systems International SARL
3a Route des Arsenaux
1700 Fribourg, Switzerland

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Purpose

The Security Target (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	Blue Coat Systems SSL Visibility Appliance EAL 3+ Augmented with ALC_FLR.3 Security Target
ST Version	Version 0.26
ST Author	Blue Coat Systems, Inc.
ST Publication Date	January 1, 2016
TOE Reference	Blue Coat Systems SSL Visibility Appliance
TOE Software Version	3.8.4FC
TOE Hardware Version	SV1800-C, SV1800-F, SV2800, SV3800

TOE developer	Blue Coat Systems, Inc.

1.3 Product Overview

The Blue Coat SSL Visibility Appliance is a high performance transparent proxy for Secure Socket Layer (SSL) network communications. It enables a variety of applications to access the plaintext (that is, the original unencrypted data) in SSL encrypted connections, and has been designed for security and network appliance manufacturers, enterprise IT organizations and system integrators. Without compromising any aspect of enterprise policies or government compliance, the SSL Visibility Appliance lets network appliances be deployed with highly granular flow analysis while maintaining line rate performance. Additionally, the SSL Visibility Appliance will not interfere with plaintext traffic transversing the network.

1.4 TOE Overview

The TOE is the Blue Coat SSL Visibility Appliance, hardware models SV1800-C, SV1800-F, SV2800, and SV3800 (hardware versions: 090-03061, 090-03062, 090-03063, and 090-03064), running Software Version: 3.8.4FC and is a hardware and software TOE. The Blue Coat SSL Visibility Appliance is an integral component to any encrypted management strategy, and offers complete visibility into encrypted traffic without requiring the re-architecting of the network infrastructure. The SSL Visibility Appliance lets you add SSL inspection capabilities to your network security architecture and close the security visibility loophole created by encrypted traffic.

The SSL Visibility Appliance provides a complete solution to the problem of dealing with threats contained within encrypted SSL traffic. A single SSL Visibility Appliance can be deployed to detect and inspect all SSL traffic that may pose a threat, and can pass the decrypted content to one or more network security appliances which can record or block any threats. The ability to feed "inspected" traffic to more than one associated security appliance ensures that SSL traffic only has to be decrypted and then re-encrypted once as it crosses the network. The SSL Visibility Appliance is designed to work alongside existing security devices such as Intrusion Prevention Systems (IPS), Intrusion Detection Systems (IDS), Data Loss Prevention systems (DLP), Network Forensic appliances, and so on. It provides a non-encrypted version of SSL traffic to the associated appliance while maintaining an end to end SSL connection between the client and server involved in the session.

Unlike most other SSL proxy devices, the SSL Visibility Appliance does not rely on the TCP destination port number being used by a session to determine if it is using SSL or not. The SSL Visibility Appliance uses deep packet inspection (DPI) to identify SSL flows. This ensures that it can find and inspect any SSL traffic in the network, even if the traffic is using non standard port numbers. The SSL Visibility Appliance incorporates flow processing hardware and cryptographic acceleration hardware, enabling it to forward non SSL traffic at multi-Gigabit/s rates, while offering industry-leading transparent proxy performance (that is, decrypting and re-encrypting) for SSL traffic.

1.4.1 TOE Environment

There are three basic connectivity modes that define how the TOE is connected to the network. These modes are identified as:

- Active-Inline
- Passive-Inline
- Passive-Tap

The Active/Passive designation refers to the associated IT environment security appliance and how it behaves. The Inline/Tap designation refers to how the TOE is connected to the network. An "Active" associated IT environment security appliance processes traffic from the TOE and then returns the traffic to the TOE, while a "Passive" appliance simply consumes traffic from the TOE.

The TOE itself can be either "Inline," or a TAP, which is connected to a network span or tap port. The following figures show the modes of operation.

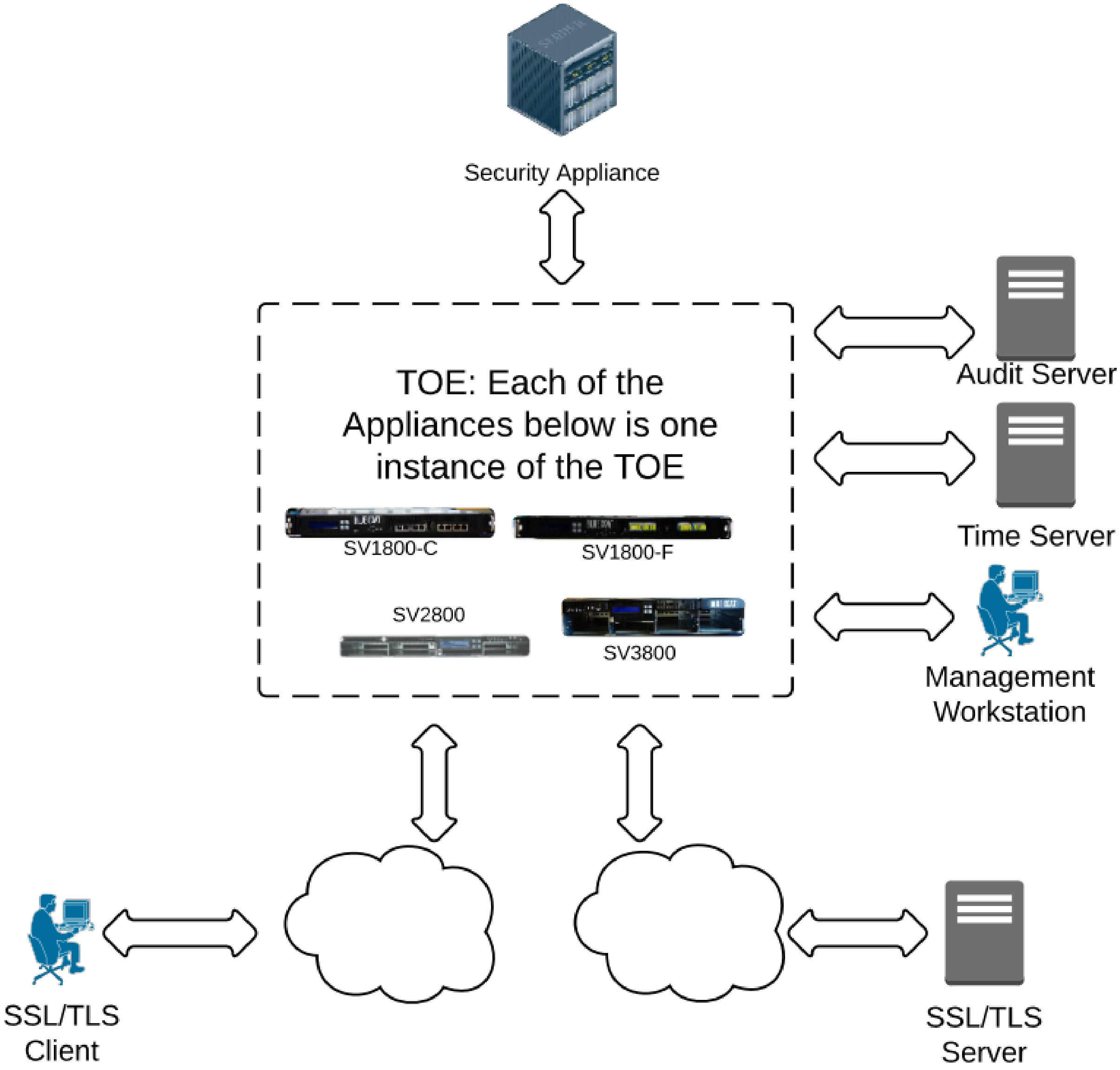


Figure 1 Passive Inline Mode

In Passive-Inline mode, network traffic flows through the TOE only, a copy of the network traffic is sent to the attached IT environment security appliance. A typical example of this type of deployment would be an IDS or Forensic appliance attached to the TOE.

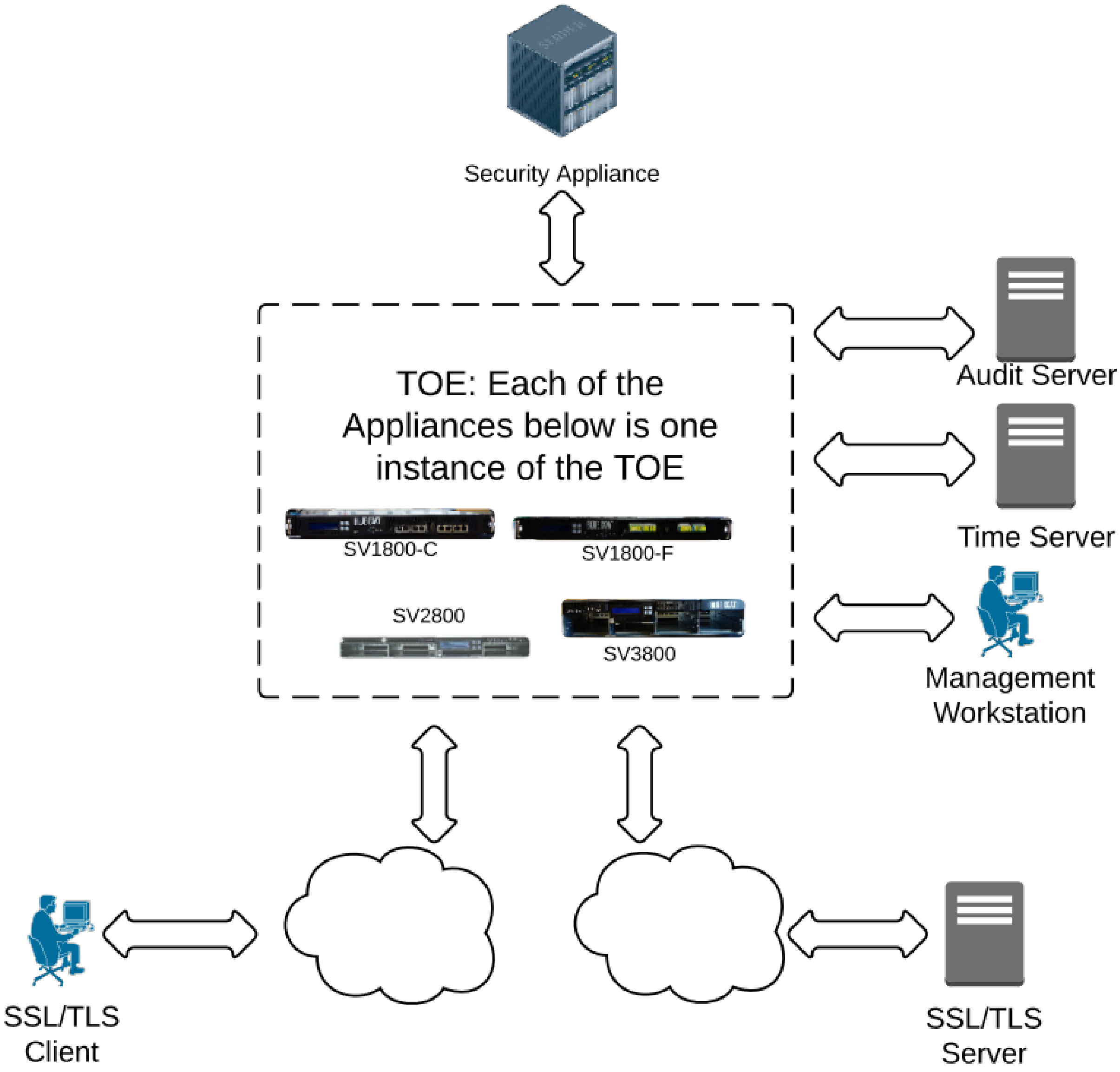


Figure 2 Active Inline Mode

In Active-Inline mode network traffic flows through both the TOE and the attached IT environment security appliance. A typical example of this type of deployment would be an IPS attached to the TOE.

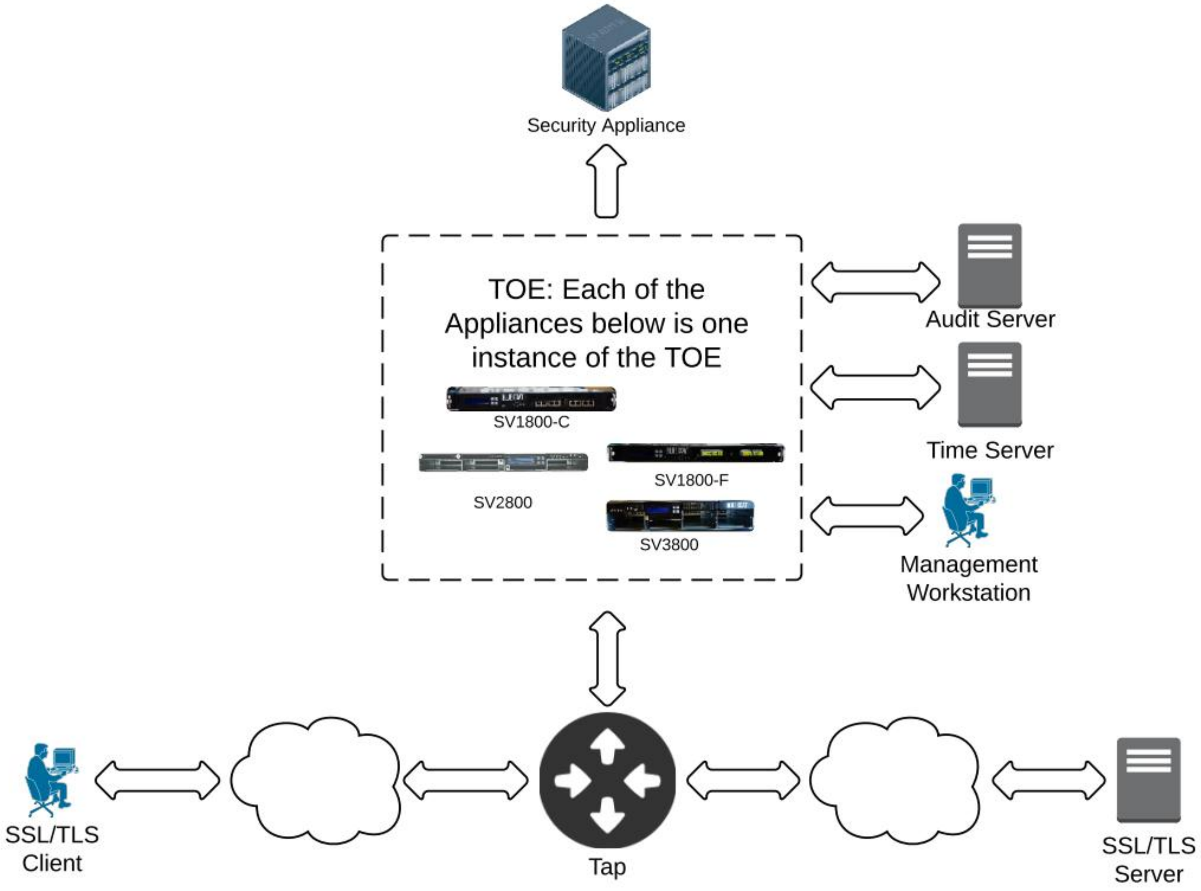


Figure 3 Passive Tap Mode

In Passive-Tap mode, network traffic does not flow through the TOE or the attached IT environment security appliance. The TOE receives a copy of traffic in the network from a TAP device and this traffic is sent to the attached IT environment security appliance. A typical example of this type of deployment would be an IDS or Forensic appliance attached to the TOE, which is in turn attached to a TAP or SPAN port.

In support of these modes of operation, the TOE supports (in some cases optionally) the following hardware, software, and firmware in its environment.

Table 2 IT Environment

Component	Required?	Usage/Purpose
Management Workstation	Yes	This workstation provides the connection to the TOE for administration and management. The Management workstation may either be directly connected to the TOE or connected over the network via a TLS or SSH connection.
Time Server	No	The TOE optionally supports the use of an NTP server for time maintenance.

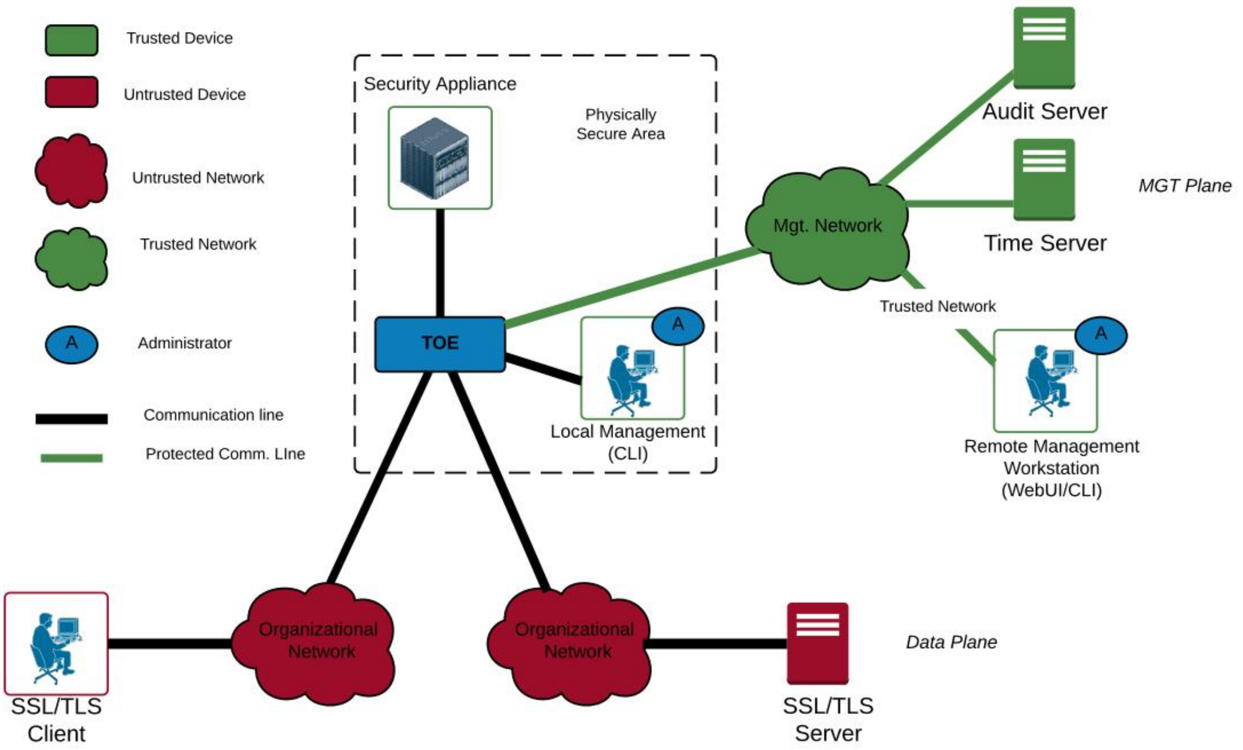
Component	Required?	Usage/Purpose
Audit Server	No	The TOE optionally sends a copy of the audit records to an external server. The connection to the audit server is a TLS-protected connection.
Security Appliance	Yes	This is the security device that is paired with the TOE to provide network protective services. Typically deployed devices may include IPS, IDS, and Network Forensic devices.
Tap	No	This IT environment device provides passive connectivity when the TOE is configured in Passive-Tap mode.
Application Servers	No	These are servers deployed on the network for which the TOE is deployed.
Network Clients	No	These are clients of the servers deployed on the network for which the TOE is deployed.
Blue Coat Network Modules (SV2800 and SV3800 only)	Yes	On the SV2800 and 3800, a network module is required to inspect SSL/TLS traffic sent to a Security Appliance and allow policy enforcement between a Client and Application Server when deployed inline. Network modules come in different configurations including 10/100/1000Base-T, 10/100/1000Base-SX, 10GBase-SR, and 10GBase-LR.

1.4.2 Protected Communications

Communications can happen in one of three ways,

1. By a management workstation directly connected to the TOE
2. On a trusted network by remote administrators, NTP servers, syslog servers, or a security appliance
3. On an untrusted network by the organization the TOE is monitoring

The following diagram provides a visual depiction of each type of communication.



Note in the figure above, the Security Appliance and Local Management workstation are in a physically secured area.

1.5 TOE Description

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

1.5.1 Physical Scope

The TOE boundary comprises the SSL Visibility Appliance and Software Version: 3.8.4FC installed on the SV1800-C, SV1800-F, SV2800, and SV3800 appliances. The TOE appliances run software that differs only in platform-specific configuration data, which describes the intended hardware platform to the OS. Differences between TOE models allow for different capacity, performance, and scalability options, as described below.

Table 3 TOE Model Comparison

	SV1800-C/F	SV2800	SV3800
Total Packet Processing Capability	8 Gbps	20 Gbps	40 Gbps
SSL Inspection Throughput	1.5 Gbps	2 Gbps	4 Gbps
Concurrent SSL Flow States	100,000	200,000	400,000
New Full Handshake SSL Sessions	7,500 per second	10,500 per second	12,500 per second

	SV1800-C/F	SV2800	SV3800
Power Supplies	1+1 Redundant 450W	1+1 Redundant 650W	1+1 Redundant 750W
Management Interfaces	2 x RJ45*	2 x RJ45*	2 x RJ45*
Dimensions (in.) HxWxD	1.75 x 17 x 20	1.75 x 17.5 x 29	3.5 x 17.5 x 29

*Note: Only 1 RJ45 interface may be configured at a time. The other interface is non-operational.

The following table identifies the SKUs (also referred to as hardware versions) associated with each hardware model.

Table 4 Hardware SKUs

Hardware Model	License	SKU/Version #
SV1800-C	Permanent License	090-03061
SV1800-F	Permanent License	090-03062
SV2800	Permanent License	090-03063
SV3800	Permanent License	090-03064

NOTE: The TOE hardware may be delivered with a version of the TOE software that is different than the certified version of the TOE. In these cases, a separate USB stick containing the certified version of the TOE will also be delivered. In these cases, the product will only be considered the TOE after the correct version of software has been installed on the hardware.

1.5.1.1 TOE Software and Hardware

The TOE is a software and hardware TOE. For the evaluated configuration, the TOE software must be installed and run on one of the following hardware configurations:

- SV1800-C
- SV1800-F
- SV2800
- SV3800

1.5.1.2 Guidance Documentation

The following guides are required reading and part of the TOE:

- Blue Coat® Systems SSL Visibility Appliance Guidance Document, 3.8.4FC, version 1.0

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 SFRs 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 administrator who is assigned one or more of the following privleges, “Manage Appliance”, “Manage Policy”, “Manage PKI”, "Auditor"accessing the TOE via the WebUI and CLD; these records are stored in the System Event Log. The TOE records the identity of the administrator responsible for the log event, where applicable. All logs can be optionally sent to a remote audit server via a mutually authenticated TLS 1.1 or 1.2 secure channel (TLS provided by the TOE's cryptographic algorithms).

1.5.2.2 Cryptographic Support

The Cryptographic Support of the TSF function provides cryptographic functions to WebUI and CLD sessions between an administrator's management workstation and the TOE. The cryptographic operations necessary to support this TSF are provided by the Blue Coat proprietary cryptographic module (Blue Coat SSL Visibility Appliance Crypto Library). Transport Layer Security (TLS) and SSH are used to secure these communications sessions. In addition, the TOE provides a variety of cryptographic algorithms for its own use including the SSL inspection functionality allowing decryption of select SSL/TLS sessions.

1.5.2.3 User Data Protection

Network packets are written into memory buffers exclusively used for packet processing. When a network packet is received by the TOE, network packets are written into 2048-bit memory buffers exclusively used for packet processing. The contents of the memory buffers include packet data and meta data. The metadata provides a mapping of packets to memory location. Packet data is not written to the areas of memory specified by the metadata as having contained packet data. Once the area of data allocated to packet data is completely used. The hardware releases the buffer for reuse and the metadata will begin pointing to the previously used sections of the buffer. Only the data that is pointed to by the metadata is used. No packet data not pointed to by the metadata is ever used. This ensures any user data that was previously present, is no longer available in the memory buffer for intentional or unintentional reuse. This guarantees that there is no residual data from the memory buffer's previous contents and therefore no potential for residual data its way into a new packet.

The TOE mediates SSL/TLS traffic into and out of a network. The TOE provides the capability to decrypt SSL connections into and out of a network. This SSL inspection capibility allows the TOE to mediate SSL/TLS traffic into and out of a network by one of several means, including,

- Allowing or disallowing SSL traffic into or out of a network.
- Passing traffic to an IT environment security appliance for analysis and policy enforcement when deployed in Active Inline Mode and Passive Inline Mode.
- Passively capturing and decrypting SSL/TLS traffic into and out of a network to be passed to IT environment appliance for offline analysis when deployed in Passive Tap Mode.

1.5.2.4 Identification and Authentication

The TOE requires administrative users to be authenticated prior to allowing access to any TOE administrative functionality. The Identification and Authentication TSF[1] 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 Administrators when they are entering their passwords at the login prompt when directly connected to the TOE via a serial connection or using a keyboard and monitor.

1.5.2.5 Security Management

The TOE provides a feature-rich WebUI and CLD for administrators to manage the security functions, configuration, and other features of the TOE. The Security Management function specifies user roles with defined access for the management of the TOE components.

[1] TSF – TOE Security Functionality

1.5.2.6 Protection of the TSF

The TOE invokes a set of self tests each time the TOE is powered on to ensure that the TSF operates correctly. The TOE implements TLS for protection of the WebUI and SSH for the protection of the CLD. TLS and SSH protect data transfer and leverages cryptographic capabilities to prevent replay attacks. 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. The TOE prevents an administrator from reading plaintext keys or passwords by encrypting this data in a secure store using the AES[2] algorithm.

1.5.2.7 TOE Access

The TOE terminates local and remote management sessions after an administrator-configurable time period of inactivity. The TOE also provides administrator's the capability to manually terminate the session prior to the inactivity timeout. After an administrator's session is terminated, the administrator must log in again to regain access to TOE functionality. A login banner is displayed for users at the login screen of the Management Console and at the login prompt of the CLD.

1.5.2.8 Trusted Path/Channels

The cryptographic functionality of the TOE provides the TOE the ability to create trusted paths and trusted channels. The TOE implements a trusted channel using TLS between itself and a remote server in order to protect the audit logs as they are being sent to the server. Additionally, the TOE provides trusted paths between administrators and the CLD via SSH, and between administrators and the WebUI via TLS/HTTPS. The management communication channels between the TOE and a remote entity are distinct from other communication channels and provide mutual identification and authentication. In addition, the communications are protected from modification and disclosure.

[2] AES – Advanced Encryption Standard

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.

2.1 Common Criteria Conformance Claim

The ST and the TOE it describes are conformant with the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Version 3.1, Revision 4, September 2012
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1, Revision 4, September 2012
    - Part 3 Conformant

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

- EAL 3 augmented with ALC_FLR.3

2.2 Protection Profile Conformance Claim

This ST and TOE it describes are not claiming conformance to any Protection Profile.

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 IT[3] 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.

Both are assumed to have a low level of motivation. The IT assets requiring protection are the TSF[4] and user data saved on the TOE. Removal, diminution and mitigation of the threats are through the objectives identified in Section 4 Security Objectives. Table 5 below lists the applicable threats.

Table 5 Threats

Name	Description
T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE 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

[3] IT – Information Technology

[4] TSF – TOE Security Functionality

Name	Description
	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.
T.SSL_HIDDEN_ATTACK	An external IT entity may circumvent established network protection mechanisms and compromise assets on a network through an uninspectable SSL/TLS tunnel.

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 6 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 6 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 7 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 7 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.
A.TRUSTED_DEVICES	The following devices connecting to the TOE are assumed to be

	trusted; NTP server (providing reliable time-stamps), Syslog Server, management workstation. These servers shall reside on a separated management network.
A.TRUSTED_SEC_APPLIANCE	The IT Environment provided Security Appliance resides within a dedicated network providing both physical and logical protection.
A.TOE_CONNECT_PHYSICAL_SEC	All equipment used to connect to the TOE, including, serial port/cable/keyboard/monitor, associated cabling/equipment, and the security appliance must remain within physically secure area at all times. This area must be collocated with the TOE itself.

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 8 Security Objectives for the TOE

Name	Description
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators and authorized IT entities, including, Audit Servers and Management Workstations.
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.
O.SSL_INSPECT	The TOE will provide the capability to decrypt SSL/TLS traffic and hand off the decrypted traffic to IT Environment security appliances for content inspection.

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 9 IT Security Objectives

Name	Description
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration, and support of the TOE.
OE.SECURITY_APPLIANCE_ENFORCEMENT	The IT Environment provides Security Appliance which will provide the TOE network traffic inspection allow/disallow decisions on decrypted SSL/TLS traffic passed from the TOE to the appliance for TOE enforcement when configured in active inline mode.
OE.TRUSTED_DEVICES	The IT environment provides trusted NTP server (providing reliable time stamps) , Management Workstations, and Syslog servers. These servers shall reside in a separated management network.

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 10 Non-IT Security Objectives

Name	Description
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.PROTECTED_TOE_CONNECT	A physically secure environment is provided for all equipment directly connecting to the TOE, including, serial port/cable/keyboard/monitor, associated cabling/equipment, and the security appliance.

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 11 identifies all extended SFRs implemented by the TOE.

Table 11 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_SSH_EXT.1	Explicit: SSH
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 extended family FAU_STG_EXT is modeled after the FAU_STG family. 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. The requirements of the extended family are focused on the secure transmission of audit records to a remote logging server.

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_EXT family.

Component Leveling

FAU_STG_EXT: Extended: Security audit event storage
1

Figure 4 Extended: Security audit event storage family decomposition

FAU_STG_EXT.1 Extended: External Audit Trail Storage is the only component of this family. This component 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. The FCS_CKM family, after which this extended family is modeled, 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. The extended family is designed to include CSP[5]s and further defines the requirements for plaintext secret and private cryptographic keys. The requirements also further define the key destruction methods allowed, per FIPS 140-2 requirements.

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_EXT family.

Component Leveling

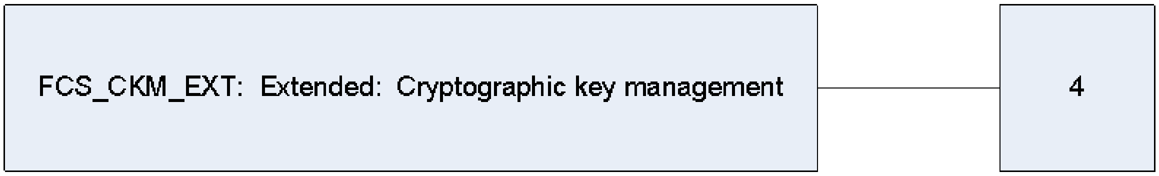


Figure 5 Extended: Cryptographic key management family decomposition

FCS_CKM_EXT.4 Extended: Cryptographic key zeroization is the only component of this family. This component requires cryptographic keys and cryptographic critical security parameters to be zeroized. It was modeled after FCS_CKM.4.

Management: FCS_CKM_EXT.4

a) There are no management activities foreseen.

Audit: FCS_CKM_EXT.4

a) There are no auditable events foreseen.

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 when no longer required.

[5] Critical Security Parameters

5.1.2.2 Family FCS_RBG_EXT: Extended: Cryptographic Operation (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

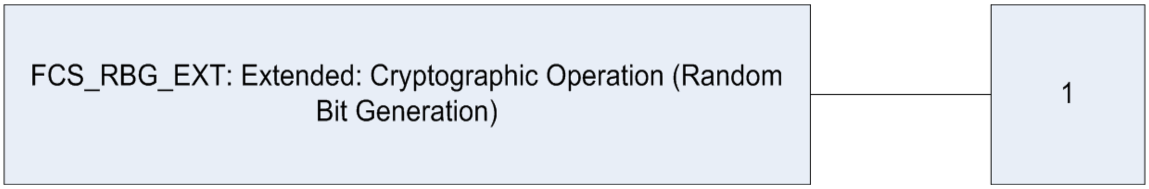

FCS_RBG_EXT: Extended: Cryptographic Operation (Random Bit Generation)
1

Figure 6 Extended: Cryptographic Operation (Random Bit Generation) family decomposition

FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation) is the only component of this class. This component 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 Cryptographic operation.

Management: FCS_RBG_EXT.1

a) There are no management activities foreseen.

Audit: FCS_RBG_EXT.1

a) There are no auditable events foreseen.

FCS_RBG_EXT.1 Extended: Cryptographic operation (Random bit generation)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RBG_EXT.1.1

The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: NIST Special Publication 800-90 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES), Dual_EC_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 security strength of the keys and hashes that it will generate.

5.1.2.3 Family FCS_TLS_EXT: Explicit: 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

FCS_TLS_EXT: Explicit: TLS
1

Figure 7 Explicit: TLS family decomposition

FCS_TLS_EXT.1 Explicit: TLS is the only component of this family. This component requires that TLS be implemented as specified.

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) Successful establishment of a TLS session.

b) Termination of a TLS session.

c) Failure to establish a TLS session.

FCS_TLS_EXT.1 Explicit: TLS

Hierarchical to: No other components.

Dependencies: FCS_COP.1(1) Cryptographic operation (for data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (for cryptographic signatures)
FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)

FCS_TLS_EXT.1.1

The TSF shall implement one or more of the following protocols [selection: 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
Optional Ciphersuites:
[selection:
None
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
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_RSA_WITH_RSA_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_RSA_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_RSA_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_RSA_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
].

5.1.2.4 Family FCS_SSH_EXT: Explicit: SSH

Family Behaviour

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

Component Leveling

Figure 8 Explicit: SSH family decomposition

FCS_SSH_EXT.1 Explicit: SSH is the only component of this family. This component requires that SSH be implemented as specified.

Management: FCS_SSH_EXT.1

a) There are no management activities foreseen.

Audit: FCS_ SSH _EXT.1

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

a) Successful establishment of an SSH session.

b) Termination of an SSH session.

c) Failure to establish an SSH session.

FCS_SSH_EXT.1 Explicit: SSH

Hierarchical to: No other components.

Dependencies: FCS_COP.1(1) Cryptographic operation (for data encryption/decryption).
FCS_COP.1(2) Cryptographic operation (for cryptographic signatures)
FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)

FCS_SSH_EXT.1.1

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: password-based.

FCS_SSH_EXT.1.2

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

FCS_SSH_EXT.1.3

The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM, *no other algorithms*].

FCS_SSH_EXT.1.4

The TSF shall ensure that data integrity algorithms used in SSH transport connection is [*selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512*].

FCS_SSH_EXT.1.5

The TSF shall ensure that diffie-hellman-group14-sha1 and [selection: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.

5.1.2.5 Family FCS_HTTPS_EXT: Explicit: 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 9 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) Successful establishment of an HTTPS session.

b) Termination of an HTTPS session.

c) Failure to establish an HTTPS session.

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[6] 2818.

FCS_HTTPS_EXT.1.2

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

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.

[6] RFC – Request For Comments

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, which is modeled after FIA_SOS Specification of secrets.

Component Leveling

Figure 10 Password Management family decomposition

FIA_PMG_EXT.1 Password Management is the only component of this family. This component 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) There are no auditable events foreseen.

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: [selection: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(”, “)”, [assignment: *other characters*]];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater.

5.1.3.2 Family FIA_UAU_EXT: 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_EXT family, which is modeled after the FIA_UAU User authentication family.

Component Leveling

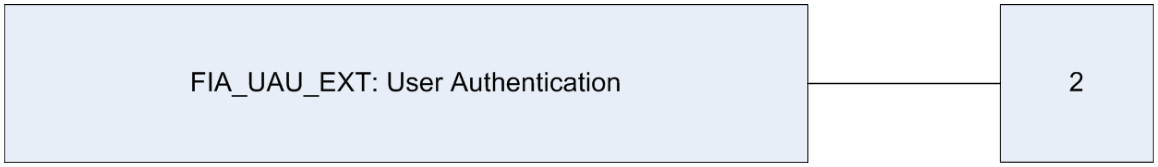


Figure 11 User authentication family decomposition

FIA_UAU_EXT.2 Extended: Password-based authentication mechanism is the only component of this family. This component requires a local password-based authentication mechanism. 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:

a) 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 administrative user authentication.

5.1.3.3 Family FIA_UIA_EXT: User Identification 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, which is modeled after the FIA_UAU and FIA_UID families.

Component Leveling

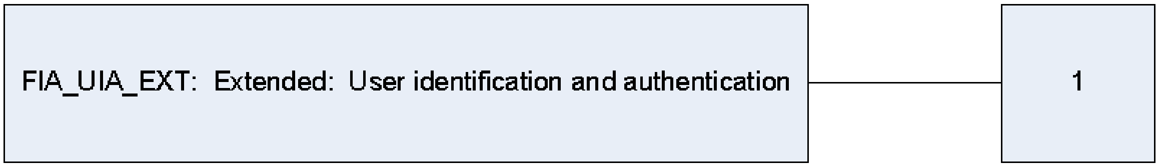

FIA_UIA_EXT: Extended: User identification and authentication
1

Figure 12 User Identification and Authentication family decomposition

FIA_UIA_EXT.1 User identification and authentication is the only component of this class, and is modeled after a combination of FIA_UAU.1 and FIA_UID.1. This component defines the actions available to users prior to initiating the identification and authentication process, and requires administrative users to be successfully identified and authenticated prior to interacting with the TSF.

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) All use of the identification and authentication mechanism.

FIA_UIA_EXT.1 User identification and authentication

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

Dependencies: FTA_TAB.1 Default TOE access banners

FIA_UIA_EXT.1.1

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

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

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_EXT: 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

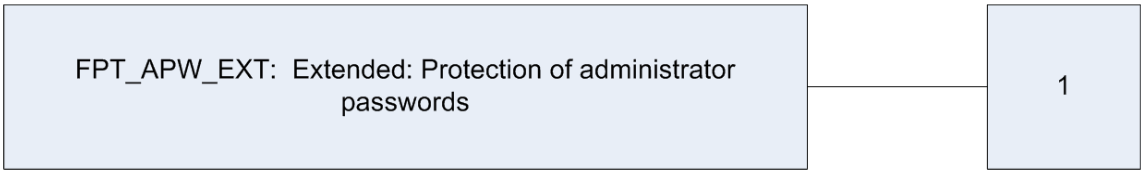

FPT_APW_EXT: Extended: Protection of administrator passwords
1

Figure 13 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.

Management: FPT_APW_EXT.1

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

a) There are no management activities foreseen.

Audit: FPT_APW_EXT.1

a) There are no auditable events foreseen.

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_APT_EXT.1.2

The TSF shall prevent the reading of plaintext passwords.

5.1.4.2 Family FPT_SKP_EXT: Extended: Protection of TSF Data

Family Behaviour

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

Component Leveling

FPT_SKP_EXT: Extended: Protection of TSF data (for reading of all symmetric keys)
1

Figure 14 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.

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_EXT: TSF Testing

Family Behaviour

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

The extended FPT_TST_EXT family is modeled after the FPT_TST family.

Component Leveling

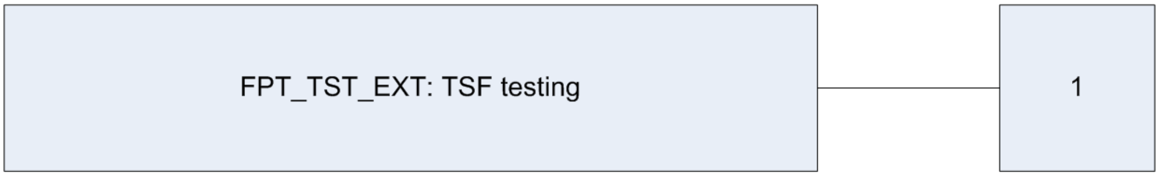


Figure 15 Extended: TSF testing family decomposition

FPT_TST _EXT.1: TSF testing is the only component of this family. This component 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_EXT: Extended: Trusted Update

Family Behaviour

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

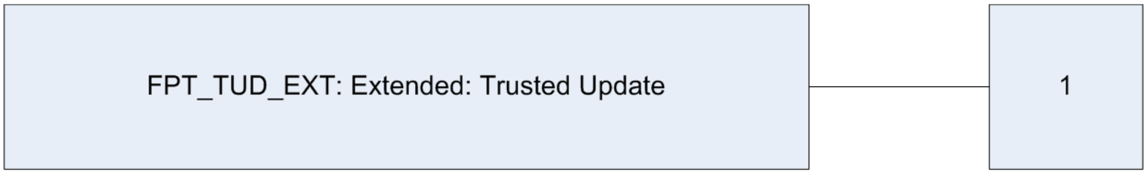


Figure 16 Extended: Trusted update family decomposition

FPT_TUD_EXT.1 Extended: Trusted update, requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation. It is the only component of this family.

Management: FPT_ TUD_EXT.1

a) There are no management activities foreseen.

Audit: FPT_ TUD_EXT.1

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

a) The initiation of the update.

FPT_TUD_EXT.1 Extended: Trusted update

Hierarchical to: No other components.

Dependencies: [FCS_COP.1(2) Cryptographic operation (for cryptographic signature), or
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

Families in this class specify functional requirements for controlling the establishment of a user's session as defined in CC Part 2.

5.1.5.1 Family FTA_SSL_EXT: TSF-initiated Session Locking

Family Behaviour

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 family is based on the FTA_SSL family.

Component Leveling

Figure 17 TSF-initiated session locking family decomposition

FTA_SSL_EXT.1: TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.

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 TSF-initiated session locking

Hierarchical to: No other components.

Dependencies: No dependencies.

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.

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:

- Assignments are shown in *italicized* text.
- Selections are shown in **bolded** text.
- Refinement are shown in *underlined italicized* text.
- Refinement deletions are shown in ~~*struckthrough italicized*~~ 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 12 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement.

Table 12 TOE Security Functional Requirements

Name	Description	S	A	R	I
FAU_GEN.1	Audit Data Generation	X	X		
FAU_GEN.2	User Identity Association				
FAU_STG_EXT.1	External Audit Trail Storage	X			
FCS_CKM.1	Cryptographic Key Generation (for Asymmetric Keys)	X		X	
FCS_CKM_EXT.4	Cryptographic Key Zeroization				
FCS_COP.1(1)	Cryptographic Operation (for Data Encryption/Decryption)	X	X	X	X
FCS_COP.1(2)	Cryptographic Operation (for Cryptographic Signature)	X		X	X
FCS_COP.1(3)	Cryptographic Operation (for Cryptographic Hashing)	X	X	X	X
FCS_COP.1(4)	Cryptographic Operation (for Keyed-Hash Message Authentication)	X	X	X	X

Name	Description	S	A	R	I
FCS_HTTPS_EXT.1	Explicit: HTTPS				
FCS_RBG_EXT.1	Extended: Cryptographic Operation (Random Bit Generation)	X			
FCS_SSH_EXT.1	Explicit: SSH	X	X		
FCS_TLS_EXT.1	Explicit: TLS	X			
FDP_IFC.1	Subset Information Flow Control		X		
FDP_IFF.1	Simple Security Attributes		X		
FDP_RIP.2	Full Residual Information Protection	X			
FIA_PMG_EXT.1	Password Management				
FIA_UAU.7	Protected Authentication Feedback		X		
FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism	X	X		
FIA_UIA_EXT.1	User Identification and Authentication	X	X		
FMT_MSA.1	Management of Security Attributes	X	X		
FMT_MSA.3	Static Attribute Initialisation	X	X		
FMT_MTD.1	Management of TSF data (for General TSF Data)	X	X		X
FMT_SMF.1	Specification of Management Functions	X	X		
FMT_SMR.2	Restrictions on Security Roles	X	X		
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_STM.1	Reliable Time Stamps				
FPT_TST_EXT.1	TSF testing				
FPT_TUD_EXT.1	Extended: Trusted Update	X			
FTA_SSL.3	TSF-initiated Termination		X	X	
FTA_SSL.4	User-initiated Termination				
FTA_SSL_EXT.1	TSF-initiated session locking	X			
FTA_TAB.1	Default TOE access banners			X	
FTP_ITC.1	Inter-TSF Trust Channel	X	X	X	
FTP_TRP.1	Trusted Path	X	X	X	

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 13.*

Table 13 Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	Start-up and shutdown of the audit functions	None.
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.
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_SSH_EXT.1	Failure to establish an SSH Session. Establishment/Termination of an SSH session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
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).
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.
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.

Requirement	Auditable Events	Additional Audit Record Contents
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 attempts.
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	Identification of the claimed user identity.

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 13***.

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** 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

The TSF shall generate *asymmetric* cryptographic keys *used for key establishment* in accordance with ~~*a specified cryptographic key generation algorithm*~~

- *NIST[7] Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;*
- *NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes*

and specified cryptographic key sizes *equivalent to, or greater than, a symmetric key strength of 112 bits* ~~*that meet the following: list of standards*~~.

FCS_CKM_EXT.4 Cryptographic key destruction

Hierarchical to: No other components.

Dependencies: FCS_CKM.1 Cryptographic key generation

FCS_CKM_EXT.4.1

The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs 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 CBC mode* and cryptographic key sizes *128-bits and 256-bits* that meets 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

The TSF shall perform *cryptographic signature services* in accordance with a ~~specified cryptographic algorithm~~ *RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater*, ~~and cryptographic key sizes~~ that meets the following:

Case: RSA Digital Signature Algorithm

- *FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"*

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

[7] NIST – National Institute of Standards and Technology

FCS_COP.1(3).1

The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm *SHA-1, SHA-224, SHA-256, SHA-384, SHA-512* and message digest ~~cryptographic key~~ sizes *160, 224, 256, 384, 512 bits* that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

Application Note: The message digests above are not applicable for each algorithm. The message digest sizes are applicable as follows, SHA-1: 160 bits, SHA-224: 224 bits, SHA-256: 256 bits, SHA-384: 384 bits, and SHA-512: 512 bits.

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

The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm *HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, key size 160, 224, 256, 384, 512 bits*, and message digest ~~cryptographic key~~ 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"*.

Application Note: The key sizes and message digests above are not applicable for each algorithm. The keys sizes and message digests are applicable as follows, HMAC-SHA1: 160 bits, HMAC-SHA224: 224 bits, HMAC-SHA256: 256 bit, HMAC-SHA384: 384 bits, and HMAC-SHA512: 512 bits.

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: No dependencies.

FCS_RBG_EXT.1.1

The TSF shall perform all random bit generation (RBG) services in accordance with **NIST Special Publication 800-90 using CTR_DRBG (AES-256)** seeded by an entropy source that accumulated entropy from **a 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 security strength of the keys and hashes that it will generate.

FCS_SSH_EXT.1 Explicit: SSH

Hierarchical to: No other components.
Dependencies: FCS_COP.1(1) Cryptographic operation (for data encryption/decryption).
FCS_COP.1(2) Cryptographic operation (for cryptographic signatures)
FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)

FCS_SSH_EXT.1.1

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: password-based.

FCS_SSH_EXT.1.2

The TSF shall ensure that, as described in RFC 4253, packets greater than **256,000** bytes in an SSH transport connection are dropped.

FCS_SSH_EXT.1.3

The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, **no other algorithms**.

FCS_SSH_EXT.1.4

The TSF shall ensure that data integrity algorithms used in SSH transport connection is **hmac-sha1**.

FCS_SSH_EXT.1.5

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

FCS_TLS_EXT.1 Explicit: TLS

Hierarchical to: No other components.

Dependencies: FCS_COP.1(1) Cryptographic operation (for data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (for cryptographic signatures)
FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)

FCS_TLS_EXT.1.1

The TSF shall implement one or more of the following protocols **TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)** supporting the following ciphersuites:

Mandatory Ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:

TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
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

6.2.3 Class FDP: User Data Protection

FDP_IFC.1 Subset Information Flow Control

Hierarchical to: No other components.
Dependencies: FDP_IFF.1 Simple Security Attributes

FDP_IFC.1.1

The TSF shall enforce the *SSL/TLS Inspection SFP* on *Subjects: SSL/TLS traffic, Information: message content, Attributes: Appliance port on which the traffic is received, SSL version, Ciphersuite, Compression Status, Certificate, Domain Name, Source IP Address, Destination IP Address, Destination Port, Traffic Class, and Operations: Drop, Reject, Allow untouched (cut through), Intercept/decrypt/reencrypt, Intercept/passively decrypt*, *external terminate signal*[8]

FDP_IFF.1 Simple Security Attributes

Hierarchical to: No other components.
Dependencies: FDP_IFC.1 Subset of information flow control.
FMT_MSA.3 Static Attribute initialisation

FDP_IFF.1.1

The TSF shall enforce the *SSL/TLS Inspection SFP* based on the following types of subject and information security attributes: *Subjects: SSL/TLS traffic, Information: traffic content, Attributes: Appliance port on which the traffic is received, SSL version, Ciphersuite, Compression Status, Certificate, Domain Name, Source IP Address, Destination IP Address, Destination Port, Traffic Class.*

FDP_IFF.1.2

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- *Traffic shall be allowed to pass untouched if any of the traffic attributes match a configured "cut through" rule.*

FDP_IFF.1.3

The TSF shall enforce the *no additional rules within the SSL/TLS Inspection SFP*.

FDP_IFF.1.4

The TSF shall explicitly authorise an information flow based on the following rules:

- *Traffic shall be intercepted, decrypted, and reencrypted if it matches a decrypt policy, has not matched a "reject" or "drop" policy, and is deployed inline or as a tap.*

FDP_IFF.1.5

The TSF shall explicitly deny an information flow based on the following rules:

- *Traffic shall be rejected if any of the traffic attributes match a configured "reject" rule,*
- *Traffic shall be dropped if any of the traffic attributes match a configured "drop rule.*
- *When working in active inline mode, traffic shall be dropped if TOE receives the terminate signal from the external security appliance*

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.

[8] The external terminate signal from the connected security appliance to drop the SSL/TLS traffic when the TOE is working in the active inline mode

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 Administrator, and support passwords of 15 characters or greater.

FIA_UIA_EXT.1 User identification and authentication

Hierarchical to: No other components.

Dependencies: FTA_TAB.1 Default TOE access banners

FIA_UIA_EXT.1.1

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

- Display the warning banner in accordance with FTA_TAB.1;
- **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.

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 user while the authentication is in progress at the local console.

6.2.5 Class FMT: Security Management

FMT_MTD.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

The TSF shall restrict the ability to ***manage*** the *TSF data* to *administrators assigned one or more of the following privileges "Manage Appliance", "Manage Policy", "Manage PKI", "Auditor".*

FMT_MSA.1 Management of Security Attributes

Hierarchical to: No other components.
Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1

The TSF shall enforce the *SSL/TLS Inspection SFP* to restrict the ability to **modify** the security attributes *Appliance port on which the traffic is received, SSL version, Ciphersuite, Compression Status, Certificate, Domain Name, Source IP Address, Destination IP Address, Destination Port, Traffic Class* to *administrator assigned the "Manage Policy" privilege..*

FMT_MSA.3 Static Attribute Initialisation

Hierarchical to: No other components.
Dependencies: FMT_MSA.1 Management of Security Attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1

The TSF shall enforce the *SSL/TLS Inspection SFP* to provide **permissive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2

The TSF shall allow the *administrator assigned the "Manage Policy" privilege* to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1 Specification of management functions

Hierarchical to: No other components.
Dependencies: No dependencies.

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;*
- *Ability to load certificates used as part of the SSL/TLS inspection functionality;*
- *Abilty to configure the SSL/TLS inspection functionality.*

FMT_SMR.2 Restrictions on security roles

Hierarchical to: FMT_SMR.1 Security roles
Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.2.1

The TSF shall maintain the roles: *administrator assigned one or more of the following privleges, "Manage Appliance", "Manage Policy", "Manage PKI", "Auditor".*

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

- *administrator assigned one or more of the following privleges, "Manage Appliance", "Manage Policy", "Manage PKI", "Auditor"shall be able to administer the TOE locally;*
- *administrator assigned one or more of the following privleges, "Manage Appliance", "Manage Policy", "Manage PKI", "Auditor"shall be able to administer the TOE remotely;*

are satisfied.

6.2.6 Class FPT: Protection of the TSF

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 key, 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 reading of the 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)]

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: No dependencies.

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

The TSF shall terminate *a remote* interactive session after a *Security 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*~~user~~-initiated termination of the *Administrator*~~user~~'s own interactive session.

FTA_TAB.1 Default TOE access banners

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_TAB.1.1

Before establishing *an administrative* user session, the TSF shall display *an Administrator-specified* advisory *notice and consent* warning message regarding ~~*unauthorized*~~ 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: No dependencies.

FTP_ITC.1.1

The TSF shall *use* **TLS** *to* provide a *trusted* communication channel between itself and *authorized IT entities supporting the following capabilities: audit server* ~~another trusted IT product~~ that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from ~~modification or~~ disclosure *and detection of modification of the channel data*.

FTP_ITC.1.2

The TSF shall permit **the TSF, or another trusted IT product** to initiate communication via the trusted channel.

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for *the audit server*.

FTP_TRP.1 Trusted path
Hierarchical to: No other components.
Dependencies: No dependencies.

FTP_TRP.1.1

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

FTP_TRP.1.2

The TSF shall permit *remote users* to initiate communication via the trusted path.

FTP_TRP.1.3

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

6.3 Security Assurance Requirements

The TOE assurance requirements for this ST are EAL3 augmented with ALC_FLR.3 derived from Common Criteria Version 3.1, Revision 4. The assurance requirements are summarized in the table below.

Table 14 TOE Assurance Requirements

Assurance Requirements	
Class ALC : Life Cycle Support	ALC_CMC.3 Authorisation controls
	ALC_CMS.3 Implementation representation CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle mode
	ALC_FLR.3 Systematic flaw remediation
Class ADV: Development	ADV_ARC.1 Security Architectural Description
	ADV_FSP.3 Functional specification with complete summary
	ADV_TDS.2 Architectural design
Class AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
Class ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: basic design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
Class AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability survey

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 15 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_SSH_EXT.1	Explicit: SSH
	FCS_TLS_EXT.1	Explicit: TLS
User Data Protection	FDP_RIP.2	Full Residual Information Protection
	FDP_IFC.1	Subset Information Flow Control
	FDP_IFF.1	Simple Security Attributes
Identification and Authentication	FIA_PMG_EXT.1	Password Management

TOE Security Function	SFR ID	Description
	FIA_UAU.7	Protected Authentication Feedback
	FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism
	FIA_UIA_EXT.1	User Identification and Authentication
Security Management	FMT_MSA.1	Management of Security Attributes
	FMT_MSA.3	Static Attribute Initialisation
	FMT_MTD.1	Management of TSF data (for General TSF Data)
	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_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 TOE's 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. (FAU_GEN.1, FAU_GEN.2)

The TOE provides auditing of all administrator actions and of all events explicitly listed in Table 13 that occur within the WebUI and CLD administrative interfaces. 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 Audit Log entries contain at a minimum the following fields:

- Date and time of the event
- Type of event
- Identity of the subject
- Outcome of the event

Additional fields will be found in addition to these fields for those events that explicitly require additional information as defined in the "Additional Audit Record Contents" column of Table 13**. (**FAU_GEN.1)

The TOE supports the SSH, TLS and HTTPS protocols and will record administrator session establishment failures, successful session establishment, and session termination events to the audit log. Session establishment failure can occur if invalid or incorrect authentication credentials are submitted.

By default, the TOE is configured to store 1GB of data before it will begin to overwrite the earliest audited events. The TOE provides the ability to securely transmit a copy of the audit logs to an external audit server using TLS. The entire Audit Log is sent encrypted using TLS to the audit server where the Audit Logs contents can be verified and viewed (FAU_STG_EXT.1). The audit logs are stored in the TOE operating system and are protected with file permissions from unauthorized access.

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

7.1.2 Cryptographic Support

Cryptographic operations necessary to support SSH, TLS, HTTPS, encryption, decryption, hashing, signature generation, signature verification, key derivation, asymmetric seed generation and key generation are provided by the TOE's Blue Coat proprietary cryptographic module (Blue Coat SSL Visibility Appliance Crypto Library) (FCS_CKM.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1, FCS_HTTPS_EXT.1, FCS_TLS_EXT.1, FCS_SSH_EXT.1). The TOE uses SSH, TLS, and HTTPS (via TLS) to protect communications. SSH provides a trusted path for remote administrators accessing the TOE's CLD. TLS is also used to provide a trusted channel during audit log transmissions from the TOE and to perform the SSL/TLS inspection functionality for the protected network. HTTPS (via TLS) is used to provide a trusted path for administrator management connections to the WebUI (FCS_HTTPS_EXT.1, FCS_TLS_EXT.1, FCS_SSH_EXT.1). The TOE uses symmetric AES keys to encrypt and decrypt data. These symmetric keys are generated/established via the TLS and SSH protocol (FCS_TLS_EXT.1, FCS_SSH_EXT.1). The TOE also provides HMAC[9]-SHA[10] and SHS[11] to support TOE cryptographic functionality (FCS_COP.1(3)).

[9] HMAC – (keyed-) Hashed Message Authentication Code

[10] SHA – Secure Hash Algorithm

[11] SHS – Secure Hash Standard

The TOE uses the following symmetric algorithms to encrypt and decrypt data, AES-128-CBC and AES-256-CBC (FCS_COP.1(1)). The TOE also uses the following MACs and hashes to support TOE cryptographic functionality HMAC[12]-SHA[13] and SHS[14] (FCS_COP.1(3)).

The TOE's cryptographic module includes self-tests for the supported FIPS-approved algorithms. For a complete list and description of the self-tests performed by the TOE, please section 7.1.6.

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. The TOE implements a CTR_DRBG (using AES-256) to generate symmetric keys and to provide seeding material to asymmetric generation functions. The TOE implements finite field cryptography (FFC) Diffie-Hellman (DH) key pair generation in accordance with section 5.6.1 of NIST Special Publication 800-56A providing at least 112 bits of key strength. The DH key pairs are used for key establishment in accordance with the FFC sections of NIST Special Publication 800-56A. The TOE implements RSA 186-3 key pair generation in accordance with section 6.3 of NIST Special Publication 800-56B providing at least 112 bits of key strength. The RSA key pair is used for key establishment in accordance with section 6.2 of NIST Special Publication 800-56B. (FCS_CKM.1)

Cryptographic keys stored internally are protected in a secure store by an AES-256 bit key encryption key (KEK).

The TOE can use AES 128 and 256-bit keys 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. TLS/HTTPS is used for management sessions via the WebUI. TLS is used to protect communications with a remote audit server.

The TOE supports the following TLS ciphersuite for administrative traffic (FCS_HTTPS_EXT.1, FCS_TLS_EXT.1):

- 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
- 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

The TOE's cryptographic module supports the following algorithms:

- AES key sizes of 128 bits and 256 bits
- AES modes of CBC
- rDSA with key sizes of 2048 bit and greater
- DH with public key sizes of 2048 bit and greater
- SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512

[12] HMAC – (keyed-) Hashed Message Authentication Code

[13] SHA – Secure Hash Algorithm

[14] SHS – Secure Hash Standard

- HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512

The TOE supports the use of the password-based for authentication over SSH. The TOE detects large SSH packets by examining the header information for incoming packets. If the packet is an SSH packet, and the packet size is greater than 256 KB, then the packet is dropped. SSH traffic can be encrypted with AES-CBC-128 and AES-CBC-256. For data integrity during SSH sessions, HMAC-SHA1 is available. Diffie-Hellman-group14-SHA1, ecdh-sha2-nistp256, and ecdh-sha2-nistp384 are the only allowed key exchange method used for the SSH protocol. (FCS_SSH_EXT.1)

The TOE provides zeroization techniques for all plaintext and private keys. TLS and SSH session keys reside in volatile memory only and never stored persistently. The contents of volatile memory are lost immediately (overwritten with zeros) when power is removed or the TOE is restarted; therefore, TLS and SSH session keys are considered zeroized when the TOE is restarted or shutdown (FCS_CKM_EXT.4).

Keys stored in the secure store can be zeroized by performing a factory default reset. When the factory default reset is performed, the key encrypting the secure store is overwritten with zeros, effectively making any encrypted information in the secure store inaccessible. Also as part of the factory default reset, the entire disk is overwritten with zeros. After the factory default reset has been triggered during the boot process, no additional commands can be given until the reset has been completed. This prevents an attacker from influencing the zeroization procedure.

Each of the TOE cryptographic algorithms have been CAVP tested. The following table identifies each of the CAVP algorithm certificates associated with each algorithm:

Table 16 Cryptographic Algorithm Certificates

Algorithm	Certificate Number
AES (FCS_COP.1(1))	3195
RSA (FCS_CKM.1, FCS_COP.1(2))	1625, 1238
ECDSA (FCS_COP.1(2))	584
SHS (FCS_COP.1(3))	2642
HMAC-SHS (FCS_COP.1(4))	2013
DRBG (FCS_RBG_EXT.1)	669

TOE Security Functional Requirements Satisfied: FCS_CKM.1, FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1, FCS_HTTPS_EXT.1, FCS_TLS_EXT.1, FCS_SSH_EXT.1.

7.1.3 User Data Protection

The TOE enforces the User Data Protection TSF on user data by ensuring that the buffer area used by previous network packets is made unavailable during the buffer allocation process. When a network packet is received by the TOE, network packets are written into 2048-bit memory buffers exclusively used for packet processing. The contents of the memory buffers include packet data and meta data. The metadata provides a mapping of packets to memory location. Packet data is not written to the areas of memory specified by the metadata as having contained packet data. Once the area of data allocated to packet data is completely used. The hardware release the buffer for reuse and the metadata will begin pointing to the previously used sections of the buffer. Only the data that is pointed to by the metadata is used. No packet data not pointed to by the metadata ever used. This ensures any user data that was previously present, is no longer available in the memory buffer for intentional or unintentional reuse. This guarantees that there is no residual data from the memory buffer’s previous contents and therefore no potential for residual data its way into a new packet. (FDP_RIP.2)

The TOE is deployed between two separate networks. When SSL/TLS traffic is sent between the two networks, the TOE captures that traffic, decrypts the traffic, and provides it to an external security appliance for further analysis (the external security appliance resides within a dedicated network which provides both physical and logical protection). Depending on the deployed configuration, the TOE can either passively capture and decrypt traffic for analysis or can be configured to actively intercept the SSL traffic, decrypt, and reencrypt the traffic to pass it to its intended destination. When the TOE is actively intercepting traffic, the external security appliance may be configured to make information flow decisions. Additionally, the TOE can make traffic flow decisions based on characteristics of the SSL/TLS connection itself when configured to actively intercept traffic (FDP_IFF.1).

SSL/TLS inspection policies may be configured and applied based on characteristics of the traffic received by the TOE. These policies can be configured based on,

- TOE port on which the traffic is received: Different policies may be applied on different ports
- SSL version: Policies may be configured to automatically drop or allow SSLv2 traffic
- Ciphersuite: Policies may be configured to automatically drop, intercept/decrypt, or allow traffic based on Ciphersuite
- Compression Status: Policies may be configured to automatically drop or allow traffic based on if compression is enabled
- Certificate: Policies may be configured to either reject, drop, intercept/decrypt, or allow traffic based on the presented certificate
- Domain Name: Policies may be configured to either reject, drop, intercept/decrypt, or allow traffic based on the traffic Domain Name
- Source IP Address: Policies may be configured to either reject, drop, intercept/decrypt, or allow traffic based on Source IP Address.
- Destination IP Address: Policies may be configured to either reject, drop, intercept/decrypt, or allow traffic based on Destination IP Address.
- Destination Port: Policies may be configured to either reject, drop, intercept/decrypt, or allow traffic based on Destination Port
- Traffic Class: Policies may be configured to either reject, drop, intercept/decrypt, or allow traffic based on Traffic Class (FDP_IFC.1)

TOE Security Functional Requirements Satisfied: FDP_RIP.2, FDP_IFC.1, FDP_IFF.1.

7.1.4 Identification and Authentication

The TOE provides mechanisms for authenticating administrators connecting to the TOE through the WebUI and CLD (FIA_UAU_EXT.2).

Administrator authentication is enforced through the use of a password. A*dministrators assigned the “Manage Appliance”privilege*can configure the password to be at least a minimum password length of fifteen (15) characters. Valid passwords can be composed of any combination of upper and lower case letters, numbers, and the following special characters: !, @, #, $, %, ^, <, &, *, (,), *comma (,), quotation mark (“), underscore (_), tab (\t), and space ( )*. (FIA_PMG_EXT.1, FIA_UAU_EXT.2)

All forms of authentication for the WebUI and CLD are secured using a trusted path or trusted channel depending on the authentication mechanism in use. The CLD only accepts credentials via a keyboard and monitor, serial connection or an SSH session. The WebUI only accepts credentials via HTTPS (over TLS) (FIA_UAU_EXT.2).

There is no feedback presented to Administrators when they are entering their passwords at the login prompt of the CLD when directly connected to the TOE via a serial connection or with a keyboard and monitor (FIA_UAU.7).

Unauthenticated users only have access to read the displayed warning banner before authenticating successfully with the TOE and establish a secure SSH or TLS session 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). The secure SSH or TLS session only provides access for the unauthenticated Administrator to authenticate and there are no other services for unauthenticated users (FIA_UIA_EXT.1).

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, administrator accounts, and information flow polcy rules. The TOE provides *administrators assigned one or more of the following privileges "Manage Appliance", "Manage Policy", "Manage PKI", "Auditor"*with the WebUI to easily manage the security functions and TSF data of the TOE. The WebUI can be used to configure the cryptographic functionality available on the TOE, update the TOE, the information flow policies and verify the updates via digital signatures (for more information on trusted updates, see section 7.1.6). The information flow policies are applied via the WebUI. Within the WebUI, the administrator can configure interfaces or groups of interfaces. For each interface grouping, the SSL/TLS inspection capabilities can be configured to either Drop, Reject, Allow untouched (cut through), Intercept/decrypt/reencrypt, or Intercept/passively decrypt. The policies are created on an attribute by attribute level via drop down boxes associated with each traffic attribute (FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_SMF.1, FMT_SMR.2).

Security management functionality is available to administrators over the CLD as well.

The TOE is managed by an administrator that may be assigned one or more of the following sets of privileges.

- "Manage Appliance" – This privilege level is meant for system and network administrators.
- "Manage Policy" – This privilege level is meant for corporate policy/compliance administrators.
- "Manage PKI" – This privilege level is meant for security officers and FIPS crypto officers.
- "Auditor" – This role is meant for auditors who monitor the events on the TOE.

Note: These privileges are available both through the GUI and CLI.

The following table identifies the services available to each of the privilege levels:

Auditor	Manage Appliance	Manage Policy	Manage PKI	Authorized Service
			Y	Unlock secure store

Y	Y	Y	Y	View dashboards
Y	Y			View system log data
		Y	Y	View/export SSL session log, SSL errors
		Y	Y	View SSL statistics
		Y	Y	View/export intercepted certificates
		Y	Y	View debug information: SSL statistics
Y	Y	Y	Y	View debug information: NFE network statistics
	Y	Y		View debug information: NSM host statistics, NSM NFP statistics
		Y		Create/edit/delete rulesets,rules, segments, and user defined lists
		Y	Y	View rulesets, rules, segments, and user defined lists
		Y		Activate/deactivatesegments.
			Y	Create/delete/export/importinternal CA keys and certificates used for re-signing
			Y	Delete/import external CA certificates
			Y	Delete/import CRLs
			Y	Import/delete trusted certificates
			Y	Import/delete known keys and certificates
		Y	Y	View PKI information
Y	Y	Y	Y	View software, hardware details
	Y			Configure appliance settings: management network, system time, alerts
Y	Y	Y	Y	View appliance settings
	Y			Configure appliance settings: remote logging configuration
	Y			Create/edit/delete user accounts
			Y	Assign/remove Manage PKI role
Y	Y		Y	View user accounts
		Y		Backup policy
		Y		Restore policy
			Y	Backup PKI information
			Y	Restore PKI information
	Y			Backup user accounts

	Y			Restore user accounts
	Y			Halt/reboot appliance
			Y	Import user interface certificate and key
		Y		Configure Host Categorization
	Y			Configure NTP Server
	Y			Update the BIOS
	Y			Update the Firmware
	Y			Configure license
Y	Y	Y	Y	Clear screen in CLI
Y	Y	Y	Y	Edit grid size in WebUI

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

7.1.6 Protection of the TSF

The TOE provides SSH, TLS, and TLS/HTTPS to protect TSF data from disclosure and to detect modification of TSF data while in transit between the TOE and external IT entities, including, the Audit Server and remote management workstations.

The TOE does not allow any Administrator to read plaintext passwords stored on the TOE, since all passwords are stored in encrypted form using an AES-256-bit key (FPT_APW_EXT.1). The TOE also prevents symmetric and private keys from being read by storing keys in encrypted form using an AES-256-bit key. The encrypting AES-256-bit key is stored in internally-allocated data structure. The TOE's OS safeguards memory and process space from unauthorized access. Because there is no direct access to memory, and passwords, private keys, and other CSPs are stored in encrypted form, there is no potential for an all-powerful Administrator to directly read plaintext CSPs from memory (FPT_SKP_EXT.1/FPT_APW_EXT.1).

The TOE generates its own time stamps that originate from a system hardware clock. Administrators may change the time through the WebUI and configure the TOE to use an NTP server. The TOE supports authenticated NTP as defined in RFC 1305 for communications with the remote NTP server. When connecting with an external NTP server, the TOE verifies an HMAC-SHA1 keyed hash over the time request response to verify that authenticity of the time from the NTP server. The key used as the HMAC key is configured by the administrator on both the TOE and the external NTP server. This NTP server is resident on a dedicated network which provides both physical and logical security and is considered trusted (FPT_STM.1).

Administrators can find the current version of TOE software by going to the home page of the WebUI or using the `version` command through the CLD. The TOE also provides a feature to update the TOE software. When a TOE software upgrade is initiated by an administrator, an integrity test public key (RSA 2048-bit SHA-256) is used to verify the digital signature of the new TOE software before it is installed. The integrity test public key resides on the TOE's hard disk. Failure to verify the integrity of the downloaded TOE software will result in an error and the administrator will be unable to proceed with the upgrade. Candidate updates are downloaded from Blue Coat's website (https://bto.bluecoat.com/download), which is the authorized source that signs these images. Access to the images requires an account with the site. All images are digitally signed by Blue Coat so they can be verified during the upgrade process (FPT_TUD_EXT.1).

At power up, the TOE runs a suite of self-tests that check for the correct operation of the cryptographic functionality provided by the TOE. All TOE appliances run these tests on startup. The TOE first performs an integrity test on the TOE software, guaranteeing that there have been no modifications, malicious or otherwise, to the TOE software. (FPT_TST_EXT.1)

The TOE proceeds to test its software implementation of cryptographic functionality through a series of known answer tests (KATs) and pairwise consistency tests, which exercise and verify the operation of the TOE's cryptographic services. Successfully completing the KATs and pairwise consistency tests provides evidence that the TOE is operating correctly. Any errors encountered during the software implementation self-tests will cause the TOE to enter a critical error state and require administrator intervention.

The TOE performs the following Power On Self Tests (POST):

- Software integrity tests check critical O/S components and appliance software binaries using RSA signature verification (2048 bit, SHA-256)
- AES encrypt/decrypt known answer tests (KAT) on software bulk ciphers (128 bit, CBC mode)
- AES encrypt/decrypt known answer tests (KAT) on software bulk ciphers (128 bit, GCM mode)
- AES encrypt/decrypt known answer tests (KAT) on software bulk ciphers (128 bit, CFB128 mode)
- Triple-DES encrypt/decrypt known answer tests (KAT) on software bulk ciphers (keying option 1)
- RSA known answer tests (KAT) on software signature operations (sign and verify) using the following digests (2048 bit PKCS#1 1.5)
    - SHA-1 (verify only)
    - SHA-224
    - SHA-256
    - SHA-384
    - SHA-512
- RSA known answer tests (KAT) on both NFPs hardware signature operations (sign and verify) using the following digests (2048 bit)
    - SHA-1 (verify only)
    - SHA-224
    - SHA-256
    - SHA-384
    - SHA-512
- RSA known answer tests (KAT) on both NFPs hardware based encryption using 2048-bit (encrypt and decrypt)
- RSA known answer tests (KAT) on software based encryption using 2048-bit (encrypt and decrypt)
- HMAC known answer tests (KAT) on software using the following digests
    - SHA-1
    - SHA-224
    - SHA-256
    - SHA-384
    - SHA-512
- SHA known answer tests (KAT) on software hash for the following

  - SHA-1
  - SHA-224
  - SHA-256
  - SHA-384
  - SHA-512
- SP 800-90A CTR DRBG known answer test (KAT)
- TRNG duplicate and zero output tests
- ECDSA known answer tests (KAT) (P-256, K-233 and SHA512)

All POSTs are run automatically at start-up. If an error is encountered, the TOE enters an error state and powers off (FPT_TST_EXT.1).

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

7.1.7 TOE Access

The TOE terminates local and remote management sessions after an Administrator configurable time period of inactivity has elapsed (FTA_SSL_EXT.1, FTA_SSL.3). Local sessions must be initiated by accessing the CLD via the serial port or using a keyboard and monitor. Remote sessions may be initiated by accessing the CLD using SSH or WebUI using HTTPS via TLS. Administrators may also terminate their sessions voluntarily (FTA_SSL.4). Users must log in again to regain access to TOE management capabilities. At the login screen Administrators are shown an advisory notice and consent warning message regarding unauthorized use of the TOE. The message is shown to users of both the WebUI and the CLD (FTA_TAB.1).

TOE Security Functional Requirements Satisfied: FTA_SSL.3, FTA_SSL.4, FTA_SSL_EXT.1, 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 CLD over SSH and the WebUI over TLS/HTTPS . The protocols and the cryptography implemented by the TOE provide adequate defense against unauthorized disclosure and provide for the detection of modification of TSF data while it is being communicated (FTP_TRP.1).

Additionally, the TOE provides a trusted channel between the TOE and the trusted IT entities used for the audit servers. The TOE protects audit log traffic by encrypting it with a secure TLS tunnel. The TLS channel prevents unauthorized disclosure and detection of modification for all audit data (FTP_ITC.1).

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.

8.1.1 Security Assurance Requirements Rationale

This Security Target claims conformance to EAL3 augmented with ALC_FLR.3. This target was chosen to ensure that the TOE has a medium level of assurance in enforcing its security functions when instantiated in its intended environment which imposes no restrictions on assumed activity on applicable networks. Additionally, the TOE has been designed to provide resistance against attackers of "Enhanced-Basic" potential.

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Blue Coat to satisfy the assurance requirements. The table below lists the details.

Table 17 Cryptographic Algorithm Certificates

Component	How the requirement will be met
ADV_ARC.1	The architecture description provides the justification how the security functional requirements are enforced, how the security features (functions) cannot be bypassed, and how the TOE protects itself from tampering by untrusted active entities. The architecture description also identifies the system initialization components and the processing that occurs when the TOE is brought into a secure state.
ADV_FSP.3	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose, method of use, parameters, parameter descriptions, and error messages. The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.
ADV_TDS.2	The TOE design describes the TOE security functional (TSF) boundary and how the TSF implements the security functional requirements. The design description includes the decomposition of the TOE into subsystems and/or modules, thus providing the purpose of the subsystem/module, the behavior of the subsystem/module and the actions the subsystem/module performs. The description also identifies the subsystem/module as SFR (security function requirement) enforcing, SFR supporting, or SFR non-interfering; thus identifying the interfaces as described in the functional specification. In addition, the TOE design describes the interactions among or between the subsystems/modules; thus providing a description of what the TOE is doing and how.
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the

Component	How the requirement will be met
	interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.3 ALC_CMS.3	The Configuration Management (CM) document(s) describes how the end user of the TOE can identify the evaluated TOE. The CM document(s), identifies the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked, how potential changes are incorporated, and the degree to which automation is used to reduce the scope for error.
ALC_DEL.1	The Delivery document describes the delivery procedures for the TOE to include the procedure on how to download certain components of the TOE from the Cisco website and how certain components of the TOE are physically delivered to the user. The delivery procedure detail how the end-user may determine if they have the TOE and if the integrity of the TOE has been maintained. Further, the delivery documentation describes how to acquire the proper license keys to use the TOE components.
ALC_DVS.1	The Lifecycle document(s) describes the security measures and controls that are in place at the development site(s), the security measures and controls that are in place regarding employees, and the security measures and controls that are in place during the development and maintenance of the TOE.
ALC_LCD.1	The Lifecycle document(s) describes the life-cycle model used to develop and maintain the TOE that includes methods, reviews, tests, and acceptance procedures.
ATE_COV.1 ATE_DPT.1 ATE_FUN.1	The Test document(s) consist of a test plan describes the test configuration, the approach to testing, and how the subsystems/modules and TSFI has been tested against its functional specification and design as described in the TOE design and the security architecture description. The test document(s) also include the test cases/procedures that show the test steps and expected results, specify the actions and parameters that were applied to the interfaces, as well as how the expected results should be verified and what they are. Actual results are also included in the set of Test documents.
ATE_IND.2	Blue Coat will provide the TOE for testing.
AVA_VAN.2	Blue Coat will provide the TOE for testing.

8.1.2 Dependency Rationale

This ST does satisfy all the requirement dependencies of the Common Criteria. Table 18 lists each requirement to which the TOE claims conformance with a dependency and indicates whether the dependent requirement was included. As Table 18 below indicates, all dependencies have been met.

Table 18 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	✓	
	FTP_ITC.1	✓	
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_EXT.4	FCS_CKM.1	✓	
FCS_COP.1(1)	FCS_CKM.1		This dependency is unresolved in NDPP
	FCS_CKM.4	✓	Although FCS_CKM.4 is not in the ST, FCS_CKM_EXT.4 provides coverage.
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		This dependency is unresolved because SHA message digests do not use keys, thus, they do not require the generation of keys.
	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,

SFR ID	Dependencies	Dependency Met	Rationale
			FCS_CKM_EXT.4 provides coverage.
	FCS_CKM.1		This dependency is unresolved in NDPP.
FCS_HTTPS_EXT.1	FCS_TLS_EXT.1	✓	
FCS_RBG_EXT.1	No dependencies	✓	
FCS_SSH_EXT.1	FCS_COP.1(1)	✓	
	FCS_COP.1(2)	✓	
	FCS_COP.1(3)	✓	
FCS_TLS_EXT.1	FCS_COP.1(1)	✓	
	FCS_COP.1(2)	✓	
	FCS_COP.1(3)	✓	
FDP_IFC.1	FDP_IFF.1	✓	
FDP_IFF.1	FDP_IFC.1	✓	
	FMT_MSA.3	✓	
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	FTA_TAB.1	✓	
FMT_MSA.1	FDP_ACC.1 or FDP_IFC.1	✓	Met be FDP_IFC.1.
	FMT_SMR.1	✓	
	FMT_SMF.1	✓	
FMT_MSA.3	FMT_MSA.1	✓	
	FMT_SMR.1	✓	
FMT_MTD.1	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_SMF.1	No dependencies	✓	

SFR ID	Dependencies	Dependency Met	Rationale
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_SKP_EXT.1	No dependencies	✓	
FPT_STM.1	No dependencies	✓	
FPT_TST_EXT.1	No dependencies	✓	
FPT_TUD_EXT.1	FCS_COP.1(2)	✓	
FTA_SSL.3	No dependencies	✓	
FTA_SSL.4	No dependencies	✓	
FTA_SSL_EXT.1	No dependencies	✓	
FTA_TAB.1	No dependencies	✓	
FTP_ITC.1	No dependencies	✓	
FTP_TRP.1	No dependencies	✓	

8.1.3 Security Objectives Rationale

The security objectives rationale shows how the security objectives correspond to assumptions, threats, and organizational security policies and provide a justification of that tracing.

The tracing shows how the security objectives O.* and OE.* trace back to assumptions A.*, threats T.*, and organizational security policies OSP.* defined by the SPD.

Table 19 OSP Tracing

	A.NO_GENERAL_PURPOSE	A.PHYSICAL	A.TRUSTED_ADMIN	A.TRUSTED_DEVICES	A.TRUSTED_SEC_APPLIANCE	A.TOE_CONNECT_PHYSICAL	T.ADMIN_ERROR	T.TSF_FAILURE	T.UNDETECTED_ACTIONS	T.UNAUTHORIZED_ACCESS	T.UNAUTHORIZED_UPDATE	T.USER_DATA_REUSE	T.SSL_HIDDEN_ATTACK	P.ACCESS_BANNER

O.PROTECTED_COMMUNICATIONS										X				
O.VERIFIABLE_UPDATES											X			
O.SYSTEM_MONITORING									X					
O.DISPLAY_BANNER														X
O.TOE_ADMINISTRATION							X							
O.RESIDUAL_INFORMATION_CLEARING												X		
O.SESSION_LOCK										X				
O.TSF_SELF_TEST								X						
O.SSL_INSPECT													X	
OE.NO_GENERAL_PURPOSE	X													
OE.SECURITY_APPLIANCE_ENFORCEMENT					X								X	
OE.PHYSICAL		X												
OE.TRUSTED_ADMIN			X											
OE.TRUSTED_DEVICES				X										
OE.PROTECTED_TOE_CONNECT					X	X								

8.1.4 Security Objectives Tracing Rationale

The justification demonstrates that the tracing of the security objectives to assumptions, threats, and OSPs is effective and all the given assumptions are upheld, all the given threats are countered, and all the given OSPs are enforced.

Table 20 OSP Tracing

Objective	Rationale
O.PROTECTED_COMMUNICATIONS	This security objective is necessary to counter the threat T.UNAUTHORIZED_ACCESS to ensure that unauthorized users do not gain access to the TOE by easdropping unprotected traffic.
O.VERIFIABLE_UPDATES	This security objective is necessary to counter the threat T.UNAUTHORIZED_UPDATE to ensure the end user has not installed a malicious update, thinking that it was legitimate.
O.SYSTEM_MONITORING	This security objective is necessary to counter the T.UNDETECTED_ACTIONS to ensure activity is monitored so the security of the TOE is not compromised.
O.DISPLAY_BANNER	This security objective is necessary to address the Organization Security Policy P.ACCESS_BANNER to ensure an advisory notice and consent warning message regarding unauthorized use of the TOE is displayed before the session is established.
O.TOE_ADMINISTRATION	This security objective is necessary to counter the

Objective	Rationale
	T.ADMIN_ERROR that ensures actions performed on the TOE are logged so that indications of a failure or compromise of a TOE security mechanism are known and corrective actions can be taken.
O.RESIDUAL_INFORMATION_CLEARING	This security objective is necessary to counter the threat T.USER_DATA_REUSE to ensure that data is not reused after it is received by the TOE.
O.SESSION_LOCK	This security objective is necessary to counter the threat: T.UNAUTHORIZED_ACCESS to ensure accounts cannot be compromised and used by an attacker that does not otherwise have access to the TOE.
O.TSF_SELF_TEST	This security objective is necessary to counter the threat T.TSF_FAILURE to ensure failure of mechanisms do not lead to a compromise in the TSF.
O.SSL_INSPECT	This security objective is necessary to counter the threat T.SSL_HIDDEN_ATTACK by ensuring that the TOE can decrypt SSL traffic for analsysis.
OE.NO_GENERAL_PURPOSE	This security objective is necessary to address the assumption A.NO_GENERAL_PURPOSE by ensuring there are no generalpurpose computing capabilities (e.g., the ability to execute arbitrary code or applications) capabilities on the TOE.
OE.SECURITY_APPLIANCE_ENFORCEMENT	This security objective is necessary to counter the threat T.SSL_HIDDEN_ATTACK by ensuring that the IT environment provides allow disallow decisions for traffic sent for processing from the TOE in active inline mode. This security objective is necessary to address the assumption A.TRUSTED_SEC_APPLIANCE by ensuring that the IT environment provides a security appliance to the TOE.
OE.PHYSICAL	This security objective is necessary to address the assumption A.PHYSICAL by ensuring the TOE and the data it contains is physically protected from unauthorized access.
OE.TRUSTED_ADMIN	This security objective is necessary to address the assumption A.TRUSTED_ADMIN by ensuring the administrators are non-hostile and follow all administrator guidance
OE.TRUSTED_DEVICES	This security objective is necessary to address the assumption A.TRUSTED_DEVICES by ensuring that the IT environment provided devices used by the TOE operate correctly and as expected.
OE.PROTECTED_TOE_CONNECT	This security objective is necessary to address the

Objective	Rationale
	assumption A.TOE_CONNECT_PHYSICAL_SEC by ensuring the equipment connecting to the TOE is physically protected from unauthorized access. This security objective is necessary to address the assumption A.TRUSTED_SEC_APPLIANCE by ensuring that the Security Appliance is both connected to the TOE and in a physically secure environment.

8.1.5 Security Functional Requirement Rationale

The security functional requirements rationale contains a tracing of SFRs to security objectives of the TOE and a set of justifications that shows that all security objectives for the TOE are effectively addressed by the SFRs

8.1.6 Tracing of SFRs to Security Objectives of the TOE

The following table shows how the SFRs trace back to security objectives of the TOE.

Table 21 Objective to SFR Mappings

	O.PROTECTED_COMMUNICATIONS	O.VERIFIABLE_UPDATES	O.SYSTEM_MONITORING	O.DISPLAY_BANNER	O.TOE_ADMINISTRATION	O.RESIDUAL_INFORMATION_CLEARING	O.SESSION_LOCK	O.TSF_SELF_TEST	O.SSL_INSPECT
Security Functional Requirements Drawn from NDPP									
FAU_GEN.1			X		X				
FAU_GEN.2			X						
FAU_STG_EXT.1	X		X		X				
FCS_CKM.1	X								
FCS_CKM_EXT.4	X								
FCS_COP.1(1)	X								
FCS_COP.1(2)	X	X							
FCS_COP.1(3)	X	X							
FCS_COP.1(4)	X								

FCS_HTTPS_EXT.1	X								
FCS_RBG_EXT.1	X								
FCS_SSH_EXT.1	X								
FCS_TLS_EXT.1	X		X						
FDP_RIP.2						X			
FIA_PMG_EXT.1					X				
FIA_UAU.7					X				
FIA_UAU_EXT.2					X				
FIA_UIA_EXT.1					X				
FMT_MTD.1					X				
FMT_SMF.1					X				
FMT_SMR.2					X				
FPT_APW_EXT.1				X					
FPT_SKP_EXT.1	X								
FPT_STM.1			X		X				
FPT_TST_EXT.1								X	
FPT_TUD_EXT.1		X							
FTA_SSL.3					X		X		
FTA_SSL.4					X		X		
FTA_SSL_EXT.1					X				
FTA_TAB.1				X					
FTP_ITC.1	X								
FTP_TRP.1	X								
Additional security requirements to support the SFRs drawn from the Part 2 CC (those that are in addition to the NDPP)									
FDP_IFC.1									X
FDP_IFF.1									X
FMT_MSA.1					X				
FMT_MSA.3					X				

8.1.7 Justification of SFR Tracing

The justification demonstrates that the SFRs address all security objectives of the TOE.

Table 22 Objective to SFR Rationale

Objective	Rationale
O.PROTECTED_COMMUNICATIONS	FAU_STG_EXT.1, FCS_CKM.1, FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_HTTPS_EXT.1, FCS_RBG_EXT.1, FCS_SSH_EXT.1, FCS_TLS_EXT.1, FPT_SKP_EXT.1, FTP_ITC.1, and FTP_TRP.1 meet this objective by ensuring the communications between the TOE and Audit Server/Management Workstation are secure by implementing the encryption protocols as defined in the SFRs and as specified by the RFCs.

O.VERIFIABLE_UPDATES	FPT_TUD_EXT.1, FCS_COP.1(2), FCS_COP.1(3) meet this objective by ensuring the update is from a trusted source, and the update can be verified by cryptographic mechanisms prior to installation.
O.SYSTEM_MONITORING	FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1, FCS_TLS_EXT.1, FPT_STM.1 meet this objective by auditing actions on the TOE. The audit records identify the user associated with the action/event, whether the action/event was successful or failed, the type of action/event, and the date/time the action/event occurred. The audit logs are transmitted securely to a remote syslog server. If connectivity to the remote syslog server is lost, the TOE will block new permit actions. The TOE will provide the *administrators assigned one or more of the following privileges "Manage Appliance", "Manage Policy", "Manage PKI", "Auditor"* the capability to review Audit data. The TOE does not have an interface to modify audit records, though there is an interface available for *administrators assigned the "Manage Appliance"* privilege to delete audit data stored locally on the TOE as provided by FAU_STG_EXT.1.
O.DISPLAY_BANNER	FTA_TAB.1 meets this objective by displaying a advisory notice and consent warning message regarding unauthorized use of the TOE.
O.TOE_ADMINISTRATION	FIA_PMG_EXT.1, FIA_UAU.7, FIA_UAU_EXT.2, FIA_UIA_EXT.1, FMT_MTD.1, FMT_SMF.1, FMT_SMR.2, FTA_SSL.3, FTA_SSL.4, FTA_SSL_EXT.1 meet this objective by ensuring the TOE supports a password-based authentication mechanism with password complexity enforcement such as, strong passwords, password life-time constraints, providing current password when changing the password, obscured password feedback when logging in, and passwords are not stored in plaintext. The TOE provides the management and configuration features to securely manage the TOE and that those functions are restricted to *administrators assigned one or more of the following privileges "Manage Appliance", "Manage Policy", "Manage PKI", "Auditor"*, and the implementation of session termination after an administrative configurable inactivity time period whereas the user must be re-authenticated. The TOE must also protect itself against attempts by unauthorized users to bypass, deactivate, or tamper with TOE

	security functions. The TOE does not have an interface to modify audit records, though there is an interface available for the authorized administrator to delete audit data stored locally on the TOE as provided by FAU_STG_EXT.1, FAU_GEN.1, and FPT_STM.1.
O.RESIDUAL_INFORMATION_CLEARING	The SFR, FDP_RIP.2 meets this objective by ensuring no left over user data from the previous transmission is included in the network traffic.
O.SESSION_LOCK	FTA_SSL_EXT.1, FTA_SSL.3 meet this objective by terminating a session due to meeting/ exceeding the inactivity time limit.
O.TSF_SELF_TEST	FPT_TST_EXT.1 meets this objective by requiring the TOE to perform self-tests upon power up.
O.SSL_INSPECT	FDP_IFC.1 and FDP_IFF.1 meet this objective by by requiring the TOE to perform SSL/TLS packet inspection and to route packets based on policies in the TOE.

9. Acronyms

This section describes the acronyms used throughout this document.

Table 23 Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CC	Common Criteria
CFB	Cipher Feedback
CLD	Command Line Diagnostics
CSP	Critical Security Parameter
CTR	Counter mode
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
EAL	Evaluation Assurance Level
ECDHE	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standard
GCM	Galois Counter Mode
HMAC	Hashed Message Authentication Code
HTTPS	Hypertext Transfer Secure Protocol
KAT	Known Answer Test
KEK	Key Encryption Key
NDPP	Network Device Protection Profile
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
PBKDF	Password Based Key Derivation Function
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
POST	Power On Self Test
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SSH	Secure Shell

Acronym	Definition
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TOE	Target of Evaluation
TRNG	True Random Number Generator
TSF	TOE Security Functionality