# Blue Coat Systems, Inc.

## Blue Coat ProxySG SG600, SG900, and SG9000 running SGOS v6.5

## Security Target

Document Version: 1.3


Prepared for:

BLUE COAT

**Blue Coat Systems, Inc.**
420 N. Mary Avenue
Sunnyvale, CA 94085
United States of America

Phone: +1 866 30-BCOAT (22628)
Email: usinfo@bluecoat.com
http://www.bluecoat.com

Prepared by:

Corsec

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

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

# Table of Contents


**1 INTRODUCTION ....... 4**
1.1 PURPOSE ....... 4
1.2 SECURITY TARGET AND TOE REFERENCES ....... 5
1.3 PRODUCT OVERVIEW ....... 6
*1.3.1 ProxySG Feature Areas ....... 6*
1.4 TOE OVERVIEW ....... 8
*1.4.1 TOE Environment ....... 9*
1.5 TOE DESCRIPTION ....... 10
*1.5.1 Physical Scope ....... 10*
*1.5.2 Logical Scope ....... 12*
*1.5.3 Product Physical/Logical Features and Functionality not included in the TOE ....... 14*
**2 CONFORMANCE CLAIMS ....... 15**
**3 SECURITY PROBLEM ....... 16**
3.1 THREATS TO SECURITY ....... 16
3.2 ORGANIZATIONAL SECURITY POLICIES ....... 17
3.3 ASSUMPTIONS ....... 17
**4 SECURITY OBJECTIVES ....... 18**
4.1 SECURITY OBJECTIVES FOR THE TOE ....... 18
4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT ....... 18
*4.2.1 IT Security Objectives ....... 18*
*4.2.2 Non-IT Security Objectives ....... 19*
**5 EXTENDED COMPONENTS ....... 20**
5.1 EXTENDED TOE SECURITY FUNCTIONAL COMPONENTS ....... 20
5.2 EXTENDED TOE SECURITY ASSURANCE COMPONENTS ....... 36
**6 SECURITY REQUIREMENTS ....... 37**
6.1 CONVENTIONS ....... 37
6.2 SECURITY FUNCTIONAL REQUIREMENTS ....... 37
*6.2.1 Class FAU: Security Audit ....... 39*
*6.2.2 Class FCS: Cryptographic Support ....... 42*
*6.2.3 Class FDP: User Data Protection ....... 45*
*6.2.4 Class FIA: Identification and Authentication ....... 46*
*6.2.5 Class FMT: Security Management ....... 47*
*6.2.6 Class FPT: Protection of the TSF ....... 48*
*6.2.7 Class FTA: TOE Access ....... 49*
*6.2.8 Class FTP: Trusted Path/Channels ....... 50*
6.3 SECURITY ASSURANCE REQUIREMENTS ....... 51
**7 TOE SUMMARY SPECIFICATION ....... 52**
7.1 TOE SECURITY FUNCTIONS ....... 52
*7.1.1 Security Audit ....... 53*
*7.1.2 Cryptographic Support ....... 54*
*7.1.3 User Data Protection ....... 56*
*7.1.4 Identification and Authentication ....... 57*
*7.1.5 Security Management ....... 58*
*7.1.6 Protection of the TSF ....... 59*
*7.1.7 TOE Access ....... 62*
*7.1.8 Trusted Path/Channels ....... 62*
**8 RATIONALE ....... 64**
8.1 CONFORMANCE CLAIMS RATIONALE ....... 64
*8.1.1 Variance Between the PP and this ST ....... 64*

*8.1.2 Security Assurance Requirements Rationale* ... *64*
*8.1.3 Dependency Rationale* ... *64*
**9 ACRONYMS AND TERMS** ... **67**
9.1 TERMINOLOGY ... 67
9.2 ACRONYMS ... 68

# Table of Figures

FIGURE 1 EVALUATED CONFIGURATION OF THE TOE ... 9
FIGURE 2 PHYSICAL TOE BOUNDARY IN THE EVALUATED CONFIGURATION ... 11
FIGURE 3 EXTENDED: SECURITY AUDIT EVENT STORAGE FAMILY DECOMPOSITION ... 21
FIGURE 4 EXTENDED: CRYPTOGRAPHIC KEY MANAGEMENT FAMILY DECOMPOSITION ... 22
FIGURE 5 EXTENDED: CRYPTOGRAPHIC OPERATION (RANDOM BIT GENERATION) FAMILY DECOMPOSITION ... 23
FIGURE 6 EXPLICIT: TLS FAMILY DECOMPOSITION ... 24
FIGURE 7 EXPLICIT: SSH FAMILY DECOMPOSITION ... 26
FIGURE 8 EXTENDED: HTTPS FAMILY DECOMPOSITION ... 27
FIGURE 9 PASSWORD MANAGEMENT FAMILY DECOMPOSITION ... 28
FIGURE 10 USER AUTHENTICATION FAMILY DECOMPOSITION ... 29
FIGURE 11 USER IDENTIFICATION AND AUTHENTICATION FAMILY DECOMPOSITION ... 30
FIGURE 12 EXTENDED: PROTECTION OF ADMINISTRATOR PASSWORDS FAMILY DECOMPOSITION ... 31
FIGURE 13 EXTENDED: PROTECTION OF TSF DATA (FOR READING OF ALL SYMMETRIC KEYS) ... 32
FIGURE 14 EXTENDED: TSF TESTING FAMILY DECOMPOSITION ... 33
FIGURE 15 EXTENDED: TRUSTED UPDATE FAMILY DECOMPOSITION ... 34
FIGURE 16 TSF-INITIATED SESSION LOCKING FAMILY DECOMPOSITION ... 35

# List of Tables

TABLE 1 ST AND TOE REFERENCES ... 5
TABLE 3 CC AND PP CONFORMANCE ... 15
TABLE 4 THREATS ... 16
TABLE 5 ORGANIZATIONAL SECURITY POLICIES ... 17
TABLE 6 ASSUMPTIONS ... 17
TABLE 7 SECURITY OBJECTIVES FOR THE TOE ... 18
TABLE 8 IT SECURITY OBJECTIVES ... 19
TABLE 9 NON-IT SECURITY OBJECTIVES ... 19
TABLE 10 EXTENDED TOE SECURITY FUNCTIONAL REQUIREMENTS ... 20
TABLE 11 TOE SECURITY FUNCTIONAL REQUIREMENTS ... 37
TABLE 12 AUDITABLE EVENTS ... 39
TABLE 13 NDPP ASSURANCE REQUIREMENTS ... 51
TABLE 14 MAPPING OF TOE SECURITY FUNCTIONS TO SECURITY FUNCTIONAL REQUIREMENTS ... 52
TABLE 15 SELF-TEST DESCRIPTIONS ... 60
TABLE 16 FUNCTIONAL REQUIREMENTS DEPENDENCIES ... 64
TABLE 17 TERMS ... 67
TABLE 18 ACRONYMS ... 68

# 1 Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the organization of the ST. The TOE is the Blue Coat ProxySG SG600, SG900, and SG9000 running SGOS[1] v6.5, and will hereafter be referred to as the TOE throughout this document. The TOE is a proprietary operating system (OS) developed specifically for use on a hardware appliance that serves as an Internet proxy and Wide Area Network (WAN) optimizer. The purpose of the appliance is to provide a layer of security between an Internal and External Network, typically an office network and the Internet, and to provide acceleration and compression of transmitted data.

## 1.1 Purpose

This ST is divided into nine sections, as follows:

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

[1] SGOS – Secure Gateway Operating System

# 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, Inc. Blue Coat ProxySG SG600, SG900, and SG9000 running SGOS v6.5 Security Target |
| **ST Version** | Version 1.3 |
| **ST Author** | Corsec Security, Inc. |
| **ST Publication Date** | 03/07/2013 |
| **TOE Reference** | Blue Coat ProxySG SG600, SG900, and SG9000 running SGOS v6.5 build: 143753 |
| **CAVP Validated Algorithm Certificates** | AES Cert. #2925<br>RSA Cert. #1534<br>SHA Cert. #2461<br>DRBG Cert. #539<br>HMAC Cert. #1852<br>Cavium AES Cert #105, #1265 |

# 1.3 Product Overview

The Product Overview provides a high-level description of the Blue Coat Blue Coat ProxySG SG600, SG900, and SG9000 running SGOS v6.5 that is the subject of the evaluation. The following section, TOE Overview, provides the introduction to the parts of the overall product offering that are specifically being evaluated.

The Blue Coat ProxySG SG600, SG900, and SG9000 running SGOS v6.5 appliances (ProxySG) is a proprietary OS and hardware appliance that together serve as an Internet proxy. The purpose of the appliance is to provide a layer of security between an Internal and External Network (typically an office network and the Internet), and to provide WAN optimization for traffic passing between networks. The ProxySG SG600, SG900, and SG9000 appliances run software that differs only in platform-specific configuration data, which describes the intended hardware platform to the OS. Differences between product models allow for different capacity, performance, and scalability options.

The security provided by the ProxySG can be used to control, protect, and monitor the Internal Network's use of controlled protocols on the External Network. The ProxySG appliances offer a choice of two "editions" via licensing: MACH5 and Proxy. The MACH5 edition appliances offer a subset of the Proxy's services and have some Proxy features disabled (as indicated below).

The controlled protocols implemented are:

- Hypertext Transfer Protocol (HTTP)
- Secure Hypertext Transfer Protocol (HTTPS)
- File Transfer Protocol (FTP)
- SOCKS[7] (not included with MACH5 edition)
- Instant Messaging (AOL[8], MSN[9]/Windows LIVE Messenger, and Yahoo!) (not included with MACH5 edition)
- Common Internet File System (CIFS)
- Real-Time Streaming Protocol (RTSP)
- Microsoft Media Streaming (MMS)
- Messaging Application Programming Interface (MAPI)
- Transmission Control Protocol (TCP) tunnelling protocols (e.g., Secure Shell (SSH), IMAP[10], POP3[11], SMTP[12])
- Telnet
- Domain Name System (DNS)

Access control is achieved by enforcing configurable policies on controlled protocol traffic to and from the Internal Network users. The policy may include authentication, authorization, content filtering, and auditing. In addition, the ProxySG provides optimization of data transfer between ProxySG nodes on a WAN using its Application Delivery Network (ADN) technology. Optimization is achieved by enforcing a configurable policy on traffic traversing the WAN.

## 1.3.1 ProxySG Feature Areas

The following paragraphs depict a brief description of the ProxySG feature areas.

[7] SOCKS – SOCKet Secure
[8] AOL – America Online
[9] MSN – The Microsoft Network
[10] IMAP – Internet Message Access Protocol
[11] POP3 – Post Office Protocol version 3
[12] SMTP – Simple Mail Transfer Protocol

#### 1.3.1.1 Administrative Access

Administrative access to the ProxySG is provided by the serial port and Ethernet port. Administrators access the serial port using a terminal emulator over a direct serial connection to the appliance. The serial port controls access to the Setup Console (used for initial configuration only) and the Command Line Interface (CLI), which is used for normal administrative operations. Administrators can also access the CLI using SSH over an Ethernet connection. Administrators access the Management Console (a Web Graphical User Interface) using HTTPS over an Ethernet connection for normal administrative operations.

#### 1.3.1.2 Security Functional Policies

After initial configuration, the ProxySG is considered operational and behaves as a proxy that either denies or allows all proxied transactions through the ProxySG. During initial configuration, the administrator must choose which policy (allow or deny) is the default. To further manage controlled protocol traffic flow, an authorised administrator defines policy rules that provide a higher level of granularity than the default accept-all or deny-all policy.

Policy rules can require authentication credentials be entered by the End User that made request. End Users are those users that make requests from within the protected Internal Network out to the External Network. End Users do not have any access to management functionality. To control access with authentication, there must be an existing list of user accounts to use for authentication. The policy rules that define what protocols will be proxied, optimized, or require authentication are expressed using Content Policy Language (CPL).

#### 1.3.1.3 Explicit and Transparent Network Environments

In order to act as a proxy and manage controlled protocol traffic between the Internal and External Network, all of the targeted traffic must flow through the appliance. Arranging for controlled protocol traffic to flow through the appliance requires configuration of the organization's network environment. There are two kinds of network deployments: explicit and transparent. In an explicit deployment, the users' client software (e.g. a web browser) is configured to access the External Network via the proxy. The client software presents the traffic to the Internal Network port of the proxy for service. In a transparent deployment, the network and proxy are configured so that the proxy can intercept controlled protocol traffic intended for the External Network. The users' software is not changed and the user may be unaware that controlled protocol traffic is passing through the proxy.

#### 1.3.1.4 Reverse Proxy

The ProxySG can function as a reverse proxy. When configured, the ProxySG can cache and deliver pictures and other non-variable content rapidly, offloading those efforts from the OCS. This frees the OCS to perform application-based services.

#### 1.3.1.5 WAN Optimization

The ProxySG provides acceleration of WAN traffic using bandwidth management, data compression, protocol optimization, and object caching[15]. The ProxySG's ADN feature, which requires two[16] ProxySGs located at each end of the WAN or LAN link, utilizes additional byte caching[17], object caching, and compression techniques to provide enhanced optimization for traffic. This decreases bandwidth usage and optimizes response time.

[15] Object caching – enables clients to retrieve previously received data from a cache, rather than across the WAN.
[16] Only a single ProxySG was tested in the CC Evaluated Configuration.
[17] Byte caching – technique in which the TOE replaces large blocks of repeated data with small tokens representing that data prior to transmission.

# 1.4 TOE Overview

The TOE Overview summarizes the usage and major security features of the TOE. The TOE Overview provides a context for the TOE evaluation by identifying the TOE type, describing the product, and defining the specific evaluated configuration.

The TOE is the Blue Coat ProxySG SG600, SG900, and SG9000 running SGOS v6.5 and is a hardware and software TOE. The purpose of the TOE is to provide a layer of security between an Internal and External Network (typically an office network and the Internet), and to provide WAN optimization for traffic passing between networks.

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 depicted below in Table 2.

**Table 2 Evaluated Platforms Comparison**

| | SG600 | SG900 | SG9000 |
|---|---|---|---|
| Concurrent Users | 450-Unlimited | 3500-Unlimited | Unlimited |
| Storage | 1x250GB – 2x250GB SATA | 2x500 GB – 4x2TB SAS | 8x500GB – 15x1TB SAS |
| Memory | 4GB | 6-32GB | 8-64GB |
| Throughput | • 2x1000 Base-T card<br>• 1x1000 Base-T card | 2x 1000 Base-T (bridged pair) | 4x1000 Base-T (1x management, 1x auxiliary, 1 bridged pair) |
| Enclosure | 1U[18] x 19” | 1U x 19” | 4U x 19” |

The TOE can be used to control, protect, and monitor the Internal Network’s use of controlled protocols on the External Network. The TOE appliances offer a choice of two “editions” via licensing: MACH5 and Proxy. The MACH5 edition appliances offer a subset of the Proxy’s services and have some Proxy features disabled (as indicated below).

The controlled protocols implemented in the evaluated configuration are:

- Hypertext Transfer Protocol (HTTP)
- Secure Hypertext Transfer Protocol (HTTPS)
- File Transfer Protocol (FTP)
- SOCKS (not included with MACH5 edition)
- Instant Messaging (AOL, MSN/Windows LIVE Messenger, and Yahoo!) (not included with MACH5 edition)
- Common Internet File System (CIFS)
- Real-Time Streaming Protocol (RTSP)
- Microsoft Media Streaming (MMS)
- Messaging Application Programming Interface (MAPI)
- Transmission Control Protocol (TCP) tunnelling protocols (e.g., Secure Shell (SSH), IMAP, POP3, SMTP)
- Telnet
- Domain Name System (DNS)

[18] U – Unit

Access control is achieved by enforcing configurable policies on controlled protocol traffic to and from the Internal Network users. The policy may include authentication, authorization, content filtering, and auditing. In addition, the ProxySG provides optimization of data transfer between ProxySG nodes on a WAN using its ADN technology. Optimization is achieved by enforcing a configurable policy on traffic traversing the WAN.

The TOE provides Administrative access via the serial port and Ethernet port. Administrators access the serial port using a terminal emulator over a direct serial connection to the appliance. The serial port controls access to the Setup Console (used for initial configuration only) and the Command Line Interface (CLI), which is used for normal administrative operations. Administrators can also access the CLI using SSH over an Ethernet connection. Administrators access the Management Console using HTTPS over an Ethernet connection for normal administrative operations.

The two primary security capabilities of the TOE are secure management of the TOE and restricting controlled protocol traffic between the Internal and External Networks. The tangible assets and management functions are protected by restricting access to administrators. Only administrators can log into the ProxySG's management interfaces, access the ProxySG's configuration, and configure policies.

Figure 1 shows the details of the evaluated configuration of the TOE.

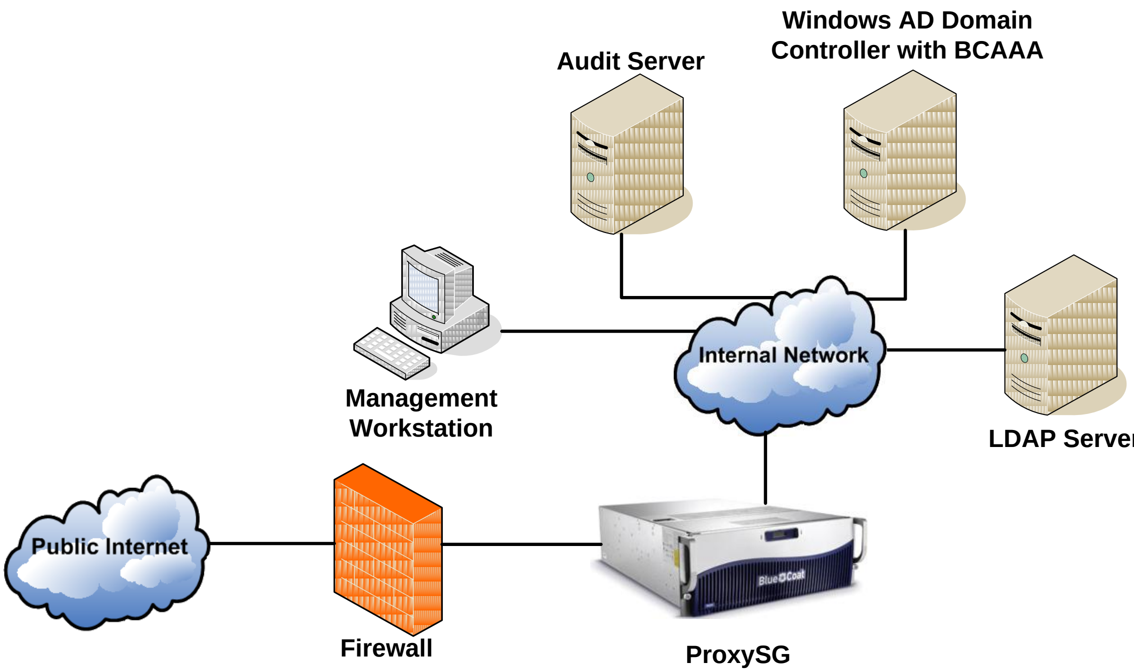


**Figure 1 Evaluated Configuration of the TOE**

### 1.4.1 TOE Environment

The TOE needs the following environmental components in order to function properly:

- cables, connectors, and switching and routing devices that allow all of the TOE and environmental components to communicate with each other
- an audit server that will contain a script to continuously pull audit logs off the TOE
- a management workstation with a standards-compliant client program to access the Management Console over HTTPS and the CLI using SSH

- a server installed with BCAAA[19] or an LDAP server for remote authentication.
- a firewall between the TOE and the External Network

The TOE is intended to be deployed in a secure data center that protects physical access to the TOE. The Audit Server, Management Workstation, and LDAP Server are all intended to be deployed in the same secure data center as the TOE. The TOE is intended to be interconnected by a back-end private network that does not connect directly to external hosts

# 1.5 TOE Description

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

## 1.5.1 Physical Scope

Figure 2 illustrates the physical scope and the physical boundary of the overall solution and ties together all of the constituents of the TOE Environment. The Blue Coat ProxySG SG600, SG900, and SG9000 running SGOS v6.5 appliances (ProxySG) is a proprietary OS and hardware appliance that together serve as an Internet proxy. The purpose of the appliance is to provide a layer of security between an Internal and External Network (typically an office network and the Internet), and to provide WAN optimization for traffic passing between networks.

The ProxySG is one of several appliances manufactured by Blue Coat Systems. The TOE appliances include the SG600, SG900, and SG9000 lines of products. All appliances run TOE software, SGOS v6.5, which differs only in platform specific configuration data, which describes the intended hardware platform to the OS.

The TOE boundary comprises the Blue Coat SG600, SG900, and SG9000 appliances and the SGOS v6.5.1.5 build 143753 software installed on the appliances.

[19] BCAAA – Blue Coat Systems Authentication and Authorization Agent – provides remote authentication over a secure channel

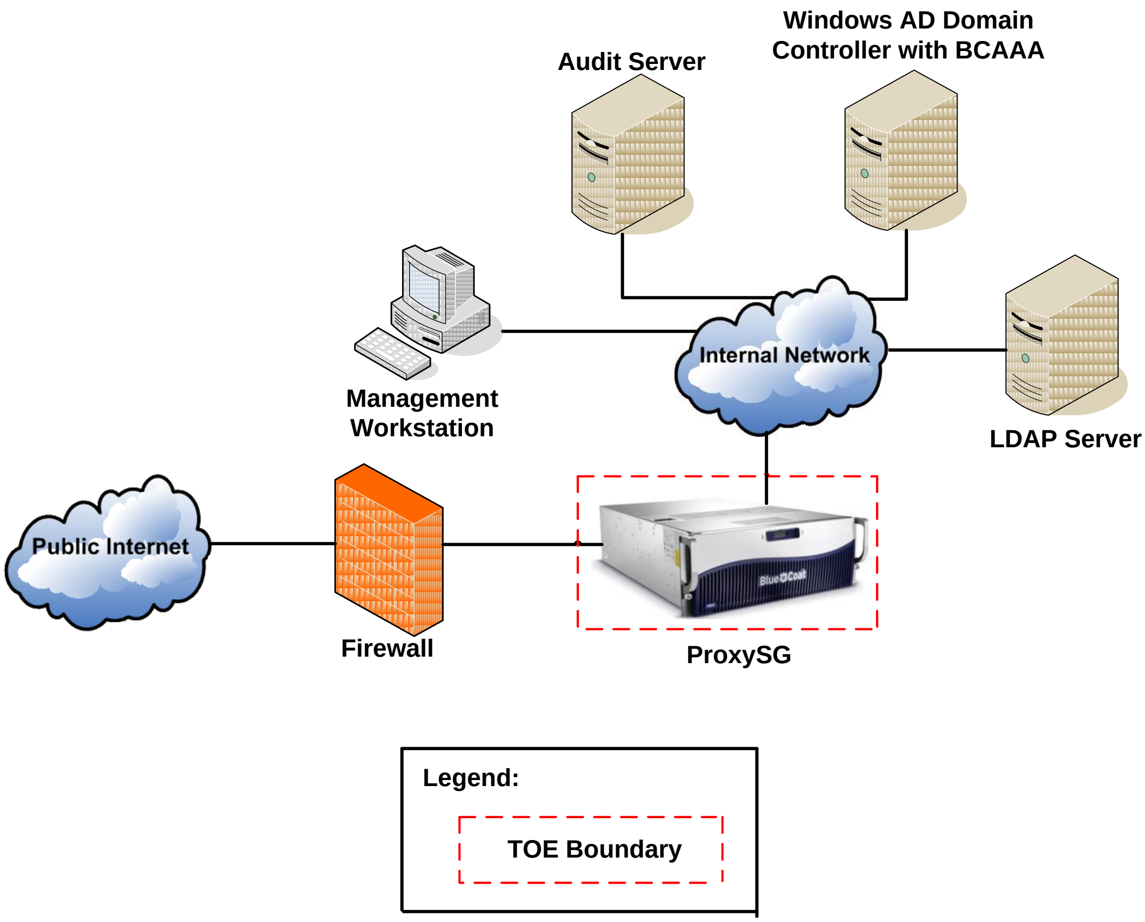


**Figure 2 Physical TOE Boundary in the Evaluated Configuration**

#### 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 Blue Coat appliance configurations:

- ProxySG SG600-10, SG600-20, SG600-35 with a Cavium CN501 Hardware Accelerator Card (HAC)
- ProxySG SG900-10, SG900-20, 900-30, SG900-45, SG900-55 with a Cavium CN1610 HAC
- ProxySG SG9000-20, SG9000-30, SG9000-40 with a Cavium CN1620 PCI-e[20] HAC

For all the above appliance models, the appliance type can be either MACH5 edition (For example, ProxySG SG600-20-M5) for WAN optimization, or Proxy edition (For example, ProxySG SG600-20-PR) for both Proxy and WAN [21]optimization features. Excluding the components listed in section 1.4.1, there are no additional hardware or environmental components are required for the TOE to function in the evaluated configuration.

[20] PCI-e – Peripheral Controller Interconnect-Express
[21] Proxy and WAN optimization features – Per the scope and SFR claims of the ND PP, only the administrative functionality was tested. No testing was performed on the Proxy or WAN Optimization features.

#### 1.5.1.2 Guidance Documentation

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

- Blue Coat Systems SGOS Administration Guide, Version SGOS 6.5.x, 231-03113, SGOS 6.5.x, 06/2014
- Blue Coat Systems Common Access Card (CAC) Solutions Guide, SGOS 6.1.2 and later, 231-03155, SGOS 6.5.x, 06/2014
- Blue Coat Notice and Consent Banner Configuration Webguide, SGOS 6.5.x, Webguide version 01-2014.03.14 (https://bto.bluecoat.com/sgos/NCB/Notice_Consent_Banner.htm)
- Blue Coat Systems SCPS[22] Deployment Guide SGOS Version 6.3.x and later, 231-03156, SGOS 6.3, 08/2013
- Blue Coat Systems ProxySG Appliance Command Line Interface Reference, version SGOS 6.5.x, 231-03035, SGOS 6.5.x, 04/2014
- Blue Coat Systems ProxySG Appliance Content Policy Language Reference, version SGOS 6.5.x, 231-03019, SGOS 6.5.3, 07/2014
- Blue Coat SGOS Upgrade/Downgrade Guide, 04/2014
- Blue Coat Systems ProxySG Appliance Visual Policy Manager Reference and Advanced Policy Tasks, SGOS Version 6.5.x, 231-03015, SGOS 6.5.x, 01/2014
- ProxySG Quick Start Guide SG600 Series, 231-03048, Rev A.0
- ProxySG Quick Start Guide: SG900 Series, 231-03109, Rev C.0
- ProxySG Quick Start Guide: SG9000 Series, 231-03159, Rev A.2
- Blue Coat ProxySG 600 Series FIPS[23] Compliance Guide: Tamper Evident Panel and Label Installation, 231-03160, C.0
- Blue Coat ProxySG 900 Series FIPS Compliance Guide: Tamper Evident Panel and Label Installation, 231-03161, B.0
- Blue Coat ProxySG 9000 Series FIPS Compliant Tamper Evident Faceplate and Label Installation Guide, 231-03063, B.0
- Blue Coat ProxySG Maintenance and Upgrade Guide, ProxySG 600 Series, 231-03051, A.2
- Blue Coat ProxySG Maintenance and Upgrade Guide, ProxySG 900 Series, 231-03166, B.1
- Blue Coat 9000 Series Maintenance & Upgrade Guide, 231-03038, E.4
- Blue Coat Systems, Inc. Blue Coat ProxySG SG600, SG900, and SG9000 running SGOS v6.5 Guidance Supplement v1.0

### 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 authorized administrators accessing the TOE via the CLI and Management Console; these records are stored in the System Event Log. The TOE records the identity of the administrator responsible for the log event, where applicable. Logs can be retrieved by an external audit server via a secure channel (HTTPS provided by the TOE's cryptographic algorithms).

#### 1.5.2.2 Cryptographic Support

The Cryptographic Support of the TSF function provides cryptographic functions to secure web browser sessions (Management Console) and terminal (CLI) sessions between an administrator's management workstation and the TOE. The cryptographic operations necessary to support this TSF are provided by the

[22] SCPS – Space Communications Protocol Specification
[23] FIPS – Federal Information Processing Standard

CAVP[24] validated Cavium CN501 (AES Cert. #105), CN1610 (AES Cert. #1265), and CN1620 (AES Cert. #1265) HACs, and Blue Coat's proprietary CAVP validated cryptographic algorithms (AES Cert. #2925, RSA Cert. #1534, SHA Cert. #2461, DRBG Cert. #539, and HMAC Cert. #1852). Transport Layer Security (TLS), HTTPS, and SSH are used to secure these communications sessions. In addition, the TOE provides a variety of cryptographic algorithms for its own use.

#### 1.5.2.3 User Data Protection

The TOE 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. Network packets are written into memory buffers used for packet processing. The size of incoming packet will be used as the size for the outgoing packet buffer allocation, ensuring any residual data in the buffer area will not make its way into the outgoing network packet. The contents of the memory buffers will be overwritten with the contents of the received packet, ensuring any user data that was previously present is no longer available in the memory buffer for intentional or unintentional reuse.

#### 1.5.2.4 Identification and Authentication

The TOE provides functionality that requires administrators to verify their claimed identity. The Identification and Authentication TSF[25] 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, X.509 certificate, or RSA key 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 of the CLI when directly connected to the TOE via a serial connection.

#### 1.5.2.5 Security Management

The TOE provides a feature-rich Management Console and a CLI 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.

#### 1.5.2.6 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 HTTPS for protection of the Management Console and SSH for the protection of the CLI. HTTPS 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 prior to storage using the AES[26] 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 CLI.

#### 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 HTTPS/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 CLI via SSH, and between administrators and the Management

[24] CAVP – Cryptographic Algorithm Validation Program
[25] TSF – TOE Security Functionality
[26] AES – Advanced Encryption Standard

Console via 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.

### 1.5.3 Product Physical/Logical Features and Functionality not included in the TOE

Features/Functionality that are not part of the evaluated configuration of the TOE are:

- ProxyClient
- BCAAA
- LDAP server
- SNMPv3 monitoring
- Remote management over Telnet
- Front panel configuration
- Remote management over HTTP
- eXtensible markup language (XML) authentication realm
- Session Monitor
- Unauthenticated access to the Visual Policy Manager (VPM)
- Unauthenticated administrative access granted via policy
- All functionality excluded from FIPS mode
- Network Time Protocol (NTP)
- Link State Propagation feature

# 2 Conformance Claims

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

**Table 3 CC and PP Conformance**

| | |
|---|---|
| **Common Criteria (CC) Identification and Conformance** | Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012; CC Part 2 extended; CC Part 3 conformant; PP claim Security Requirements for Network Devices Protection Profile conformant; Parts 2 and 3 Interpretations of the CEM[27] as of 2013/03/07 were reviewed, and no interpretations apply to the claims made in this ST. |
| **PP Identification** | Exact Conformance[28] to Security Requirements for Network Devices v1.1 (NDPP) |

[27] CEM – Common Evaluation Methodology

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

# 3 Security Problem

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

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

## 3.1 Threats to Security

This section identifies the threats to the IT[29] 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[30] 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 4 below lists the applicable threats.

**Table 4 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 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. |

[29] IT – Information Technology
[30] TSF – TOE Security Functionality

## 3.2 Organizational Security Policies

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

**Table 5 Organizational Security Policies**

| Name | Description |
|---|---|
| P.ACCESS_BANNER | The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE. |

## 3.3 Assumptions

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

**Table 6 Assumptions**

| Name | Description |
|---|---|
| A.NO_GENERAL_PURPOSE | It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. |
| A.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment. |
| A.TRUSTED_ADMIN | TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner. |

# 4 Security Objectives

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

## 4.1 Security Objectives for the TOE

The specific security objectives for the TOE are as follows:

**Table 7 Security Objectives for the TOE**

| Name | Description |
|---|---|
| O.PROTECTED_COMMUNICATIONS | The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities. |
| O.VERIFIABLE_UPDATES | The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the Administrator to be unaltered and (optionally) from a trusted source. |
| O.SYSTEM_MONITORING | The TOE will provide the capability to generate audit data and send those data to an external IT entity. |
| O.DISPLAY_BANNER | The TOE will display an advisory warning regarding use of the TOE. |
| O.TOE_ADMINISTRATION | The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators. |
| O.RESIDUAL_INFORMATION_CLEARING | The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated. |
| O.SESSION_LOCK | The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked. |
| O.TSF_SELF_TEST | The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly. |

## 4.2 Security Objectives for the Operational Environment

### 4.2.1 IT Security Objectives

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

**Table 8 IT Security Objectives**

| Name | Description |
|---|---|
| OE.NO_GENERAL_PURPOSE | There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. |

### 4.2.2 Non-IT Security Objectives

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

**Table 9 Non-IT Security Objectives**

| Name | Description |
|---|---|
| OE.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. |

# 5 Extended Components

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

## 5.1 Extended TOE Security Functional Components

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

**Table 10 Extended TOE Security Functional Requirements**

| Name | Description |
|---|---|
| FAU_STG_EXT.1 | External Audit Trail Storage |
| FCS_CKM_EXT.4 | Cryptographic key destruction |
| FCS_HTTPS_EXT.1 | Explicit: HTTPS |
| FCS_RBG_EXT.1 | Extended: Cryptographic Operation (Random Bit Generation) |
| FCS_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

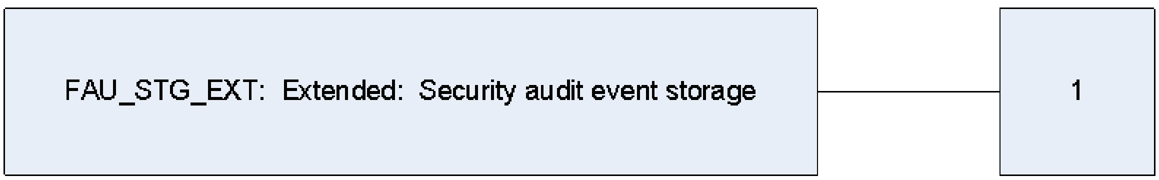


**Figure 3 Extended: Security audit event storage family decomposition**

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[31]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

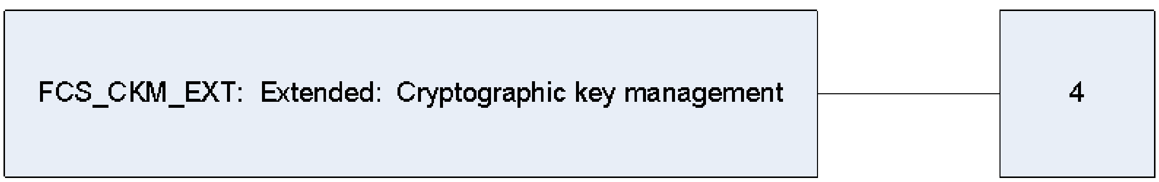


**Figure 4 Extended: Cryptographic key management family decomposition**

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.

[31] 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

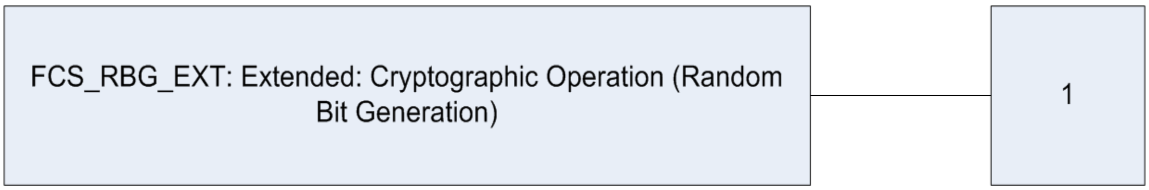


**Figure 5 Extended: Cryptographic Operation (Random Bit Generation) family decomposition**

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 bit length of the keys and authorization factors 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


**Figure 6 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
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
**Optional Ciphersuites:**
[selection:
None
*TLS_RSA_WITH_AES_128_CBC_SHA256*
*TLS_RSA_WITH_AES_256_CBC_SHA256*
*TLS_DHE_RSA_WITH_AES_128_CBC_SHA256*
*TLS_DHE_RSA_WITH_AES_256_CBC_SHA256*
*TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256*
*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 7 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_SSH_EXT.1.1***
The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, and 4254.

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

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

***FCS_SSH_EXT.1.4***
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.1.5***
The TSF shall ensure that the SSH transport implementation uses SSH_RSA and [selection: *PGP-SIGN-RSA, PGP-SIGN-DSS, no other public key algorithms*] as its public key algorithm(s).

***FCS_SSH_EXT.1.6***
The TSF shall ensure that data integrity algorithms used in SSH transport connection is [hmac-sha1, hmac-sha1-96, hmac-md5, hmac-md5-96].

***FCS_SSH_EXT.1.7***

The TSF shall ensure that diffie-hellman-group14-sha1 is the only allowed key exchange method 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 8 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[32] 2818.
***FCS_HTTPS_EXT.1.2***
The TSF shall implement the HTTPS protocol using TLS as specified in FCS_TLS_EXT.1.

[32] RFC – Request For Comments

## 5.1.3 Class FIA: Identification and Authentication

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

### 5.1.3.1 Family FIA_PMG_EXT: Password Management

Family Behaviour

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

This section defines the extended components for the FIA_PMG_EXT family, which is modeled after FIA_SOS Specification of secrets.

Component Leveling


**Figure 9 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

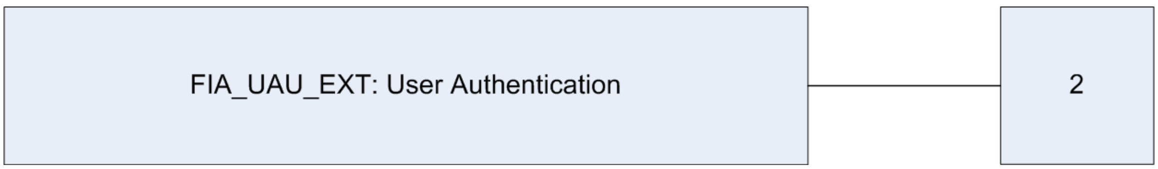


**Figure 10 User authentication family decomposition**

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

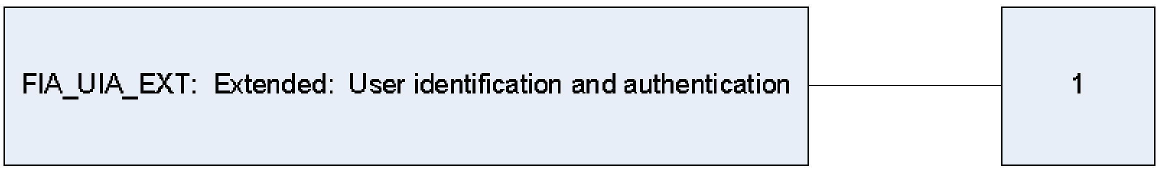


**Figure 11 User Identification and Authentication family decomposition**

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

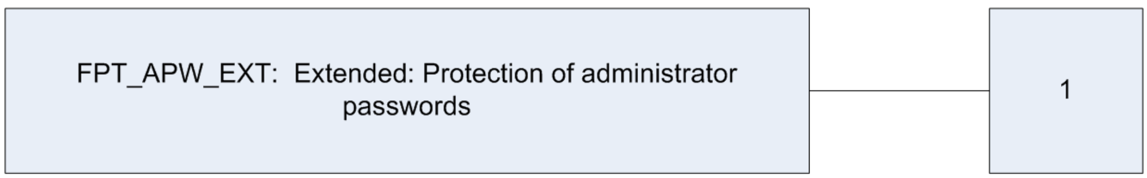


**Figure 12 Extended: Protection of administrator passwords family decomposition**

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

Management: FPT_APW_EXT.1

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

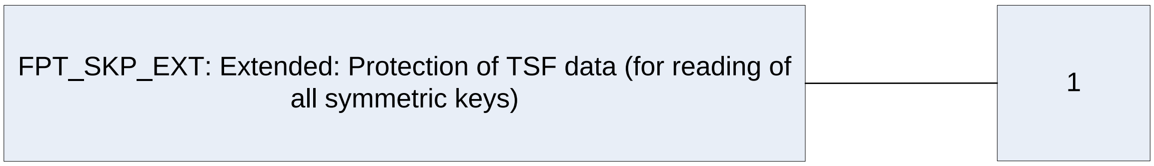


**Figure 13 Extended: Protection of TSF data (for reading of all symmetric keys)**

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

Management: FPT_SKP_EXT.1

a) There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

a) There are no audit activities foreseen.

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

#### 5.1.4.3 Family FPT_TST_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

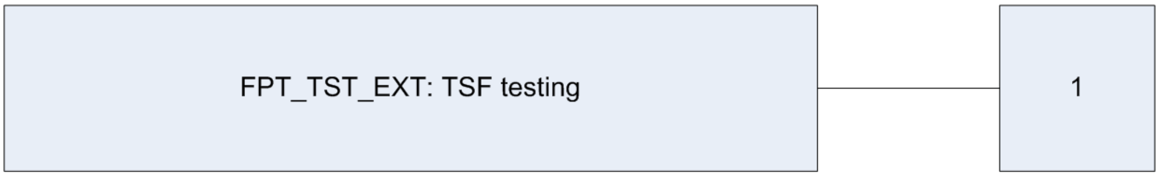


**Figure 14 Extended: TSF testing family decomposition**

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

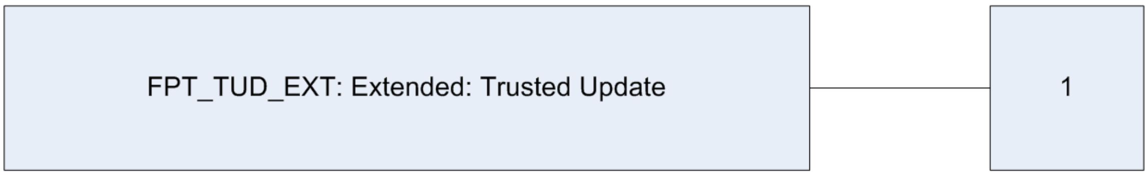


**Figure 15 Extended: Trusted update family decomposition**

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 16 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 to aid the Security Target reader.

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

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

## 6.2 Security Functional Requirements

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

**Table 11 TOE Security Functional Requirements**

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

| Name | Description | S | A | R | I |
|---|---|---|---|---|---|
| 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 | ✓ | | | |
| FDP_RIP.2 | Full Residual Information Protection | ✓ | | | |
| FIA_PMG_EXT.1 | Password Management | | | | |
| FIA_UAU.7 | Protected Authentication Feedback | | ✓ | | |
| FIA_UAU_EXT.2 | Extended: Password-based Authentication Mechanism | ✓ | ✓ | | |
| FIA_UIA_EXT.1 | User Identification and Authentication | ✓ | ✓ | | |
| FMT_MTD.1 | Management of TSF data (for General TSF Data) | ✓ | ✓ | | ✓ |
| FMT_SMF.1 | Specification of Management Functions | ✓ | ✓ | | |
| FMT_SMR.2 | Restrictions on Security Roles | ✓ | ✓ | | |
| 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 | ✓ | | | |
| FTA_SSL.3 | TSF-initiated Termination | | ✓ | ✓ | |
| FTA_SSL.4 | User-initiated Termination | | | | |
| FTA_SSL_EXT.1 | TSF-initiated session locking | ✓ | | | |
| FTA_TAB.1 | Default TOE access banners | | | ✓ | |
| FTP_ITC.1 | Inter-TSF Trust Channel | ✓ | ✓ | ✓ | |
| FTP_TRP.1 | Trusted Path | ✓ | ✓ | ✓ | |

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

## 6.2.1 Class FAU: Security Audit

**FAU_GEN.1 Audit data generation**
**Hierarchical to: No other components.**
**Dependencies: FPT_STM.1 Reliable time stamps**
***FAU_GEN.1.1***

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

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

**Table 12 Auditable Events**

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

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FIA_UAU_EXT.2 | All use of the authentication mechanism. | Origin of the attempt (e.g., IP address). |
| FIA_UAU.7 | None. | None. |
| FMT_MTD.1 | None. | None. |
| FMT_SMF.1 | None. | None. |
| FMT_SMR.2 | None. | None. |
| FPT_APW_EXT.1 | None. | None. |
| FPT_SKP_EXT.1 | None. | None. |
| FPT_STM.1 | Changes to the time. | The old and new values for the time.<br>Origin of the attempt (e.g., IP address). |
| FPT_TUD_EXT.1 | Initiation of update. | No additional information. |
| FPT_TST_EXT.1 | None. | None. |
| FTA_SSL_EXT.1 | Any attempts at unlocking of an interactive session. | No additional information. |
| FTA_SSL.3 | The termination of a remote session by the session locking mechanism. | No additional information. |
| FTA_SSL.4 | The termination of an interactive session. | No additional information. |
| FTA_TAB.1 | None. | None. |
| FTP_ITC.1 | Initiation of the trusted channel.<br>Termination of the trusted channel.<br>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.<br>Termination of the trusted path.<br>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 12*].

**FAU_GEN.2 User identity association**

**Hierarchical to: No other components.**

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

***FAU_GEN.2.1***

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

**FAU_STG_EXT.1 External audit trail storage**
**Hierarchical to: No other components.**
**Dependencies: FAU_GEN.1 Audit data generation**
**FTP_ITC.1 Inter-TSF trusted channel**

***FAU_STG_EXT.1.1***

The TSF shall be able to [transmit the generated audit data to an external IT entity] using a trusted channel implementing the [TLS/HTTPS] protocol.

## 6.2.2 Class FCS: Cryptographic Support

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

***FCS_CKM.1.1***

**Refinement:** The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance [*NIST*[33] *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*.

**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***

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

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

[33] NIST – National Institute of Standards and Technology

### FCS_COP.1(2) Cryptographic operation (for cryptographic signature)

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

***FCS_COP.1(2).1***

**Refinement:** The TSF shall perform **cryptographic signature services** in accordance with a **[*RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater*]** that meets the following:

- **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**

***FCS_COP.1(3).1***

**Refinement:** 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 sizes** [**160, 224, 256, 384, 512**] **bits** that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

### FCS_COP.1(4) Cryptographic operation (for keyed-hash message authentication)

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

***FCS_COP.1(4).1***

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

### FCS_HTTPS_EXT.1 Explicit: HTTPS

**Hierarchical to: No other components.**
**Dependencies: FCS_TLS_EXT.1**

***FCS_HTTPS_EXT.1.1***

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

***FCS_HTTPS_EXT.1.2***

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

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

**Hierarchical to: No other components.**
**Dependencies: 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 (using AES-256)]] seeded by an entropy source that accumulated entropy from [a software-based noise source; a TSF-hardware-based noise source].

***FCS_RBG_EXT.1.2***

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

### FCS_SSH_EXT.1 Explicit: SSH

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

***FCS_SSH_EXT.1.1***
The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, and 4254.

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

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

***FCS_SSH_EXT.1.4***
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.1.5***
The TSF shall ensure that the SSH transport implementation uses SSH_RSA and [*no other public key algorithms*] as its public key algorithm(s).

***FCS_SSH_EXT.1.6***
The TSF shall ensure that data integrity algorithms used in SSH transport connection is [*hmac-sha1, hmac-sha1-96*].

***FCS_SSH_EXT.1.7***
The TSF shall ensure that diffie-hellman-group14-sha1 is the only allowed key exchange method 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 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

**Mandatory Ciphersuites:**
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA

**Optional Ciphersuites:**
[*TLS_RSA_WITH_AES_128_CBC_SHA256*
*TLS_RSA_WITH_AES_256_CBC_SHA256*].

### 6.2.3 Class FDP: User Data Protection

**FDP_RIP.2 Full Residual Information Protection**
**Hierarchical to: No other components.**
**Dependencies: No dependencies.**
***FDP_RIP.2.1***

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

## 6.2.4 Class FIA: Identification and Authentication

**FIA_PMG_EXT.1 Password management**
**Hierarchical to: No other components.**
**Dependencies: No dependencies.**
***FIA_PMG_EXT.1.1***

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

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(”, “)”, [*all other ASCII*[34] *characters including ‘ ‘ (the space character)*]];
2. Minimum password length shall settable by the Security 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, [[*local RSA*[35] *public key-based authentication mechanism, X.509 certificate-based authentication mechanism, Integrated Windows Authentication (IWA) realm using BCAAA or LDAP*]] 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.

[34] ASCII – American Standard Code for Information Interchange
[35] RSA – Rivest, Shamir, Adleman

### 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 the *Security Administrators*[36].

**FMT_SMF.1 Specification of management functions**
**Hierarchical to: No other components.**
**Dependencies: FIA_UIA_EXT.1 User Identification and Authentication**
**FCS_COP.1(2) Cryptographic operation (for cryptographic signature)**
**FPT_TUD_EXT.1 Extended: Trusted Update**

***FMT_SMF.1.1***

The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;
- Ability to update the TOE, and to verify the updates using [*digital signature*] capability prior to installing those updates;
- [*Ability to configure the cryptographic functionality*]

**FMT_SMR.2 Restrictions on security roles**
**Hierarchical to: No other components.**
**Dependencies: FIA_UID.1 Timing of identification**

***FMT_SMR.2.1***

The TSF shall maintain the roles:

- **Authorized Administrator**.

***FMT_SMR.2.2***

The TSF shall be able to associate users with roles.

***FMT_SMR.2.3***

The TSF shall ensure that the conditions

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

are satisfied.

[36] Security Administrators are administrators of the TOE with the Privileged Administrator role.

## 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***

**Refinement:** 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-initiated termination of the Administrator's own interactive session.

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

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

## 6.2.8 Class FTP: Trusted Path/Channels

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

***FTP_ITC.1.1***

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

***FTP_ITC.1.2***

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

***FTP_ITC.1.3***

The TSF shall initiate communication via the trusted channel for [*the remote authentication server*].

**FTP_TRP.1 Trusted path**
**Hierarchical to: No other components.**
**Dependencies: FCS_SSH_EXT.1 Explicit: SSH**
**FCS_HTTPS_EXT.1 Explicit: HTTPS**
**FCS_TLS_EXT.1 Explicit: TLS**

***FTP_TRP.1.1***

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

***FTP_TRP.1.2***

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

***FTP_TRP.1.3***

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

# 6.3 Security Assurance Requirements

This section defines the assurance requirements for the TOE. Assurance requirements are taken from the CC Part 3.

Table 13 below summarizes the requirements.

**Table 13 NDPP Assurance Requirements**

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

# 7 TOE Summary Specification

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

## 7.1 TOE Security Functions

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

**Table 14 Mapping of TOE Security Functions to Security Functional Requirements**

| TOE Security Function | SFR ID | Description |
|---|---|---|
| Security Audit | FAU_GEN.1 | Audit Data Generation |
| | FAU_GEN.2 | User Identity Association |
| | FAU_STG_EXT.1 | External Audit Trail Storage |
| Cryptographic Support | FCS_CKM.1 | Cryptographic Key Generation (for Asymmetric Keys) |
| | FCS_CKM_EXT.4 | Cryptographic Key Zeroization |
| | FCS_COP.1(1) | Cryptographic Operation (for Data Encryption/Decryption) |
| | FCS_COP.1(2) | Cryptographic Operation (for Cryptographic Signature) |
| | FCS_COP.1(3) | Cryptographic Operation (for Cryptographic Hashing) |
| | FCS_COP.1(4) | Cryptographic Operation (for Keyed-Hash Message Authentication) |
| | FCS_HTTPS_EXT.1 | Explicit: HTTPS |
| | FCS_RBG_EXT.1 | Extended: Cryptographic Operation (Random Bit Generation) |
| | FCS_SSH_EXT.1 | Explicit: SSH |
| | FCS_TLS_EXT.1 | Explicit: TLS |
| User Data Protection | FDP_RIP.2 | Full Residual Information Protection |
| Identification and Authentication | FIA_PMG_EXT.1 | Password Management |
| | FIA_UAU.7 | Protected Authentication Feedback |
| | FIA_UAU_EXT.2 | Extended: Password-based Authentication Mechanism |

| TOE Security Function | SFR ID | Description |
|---|---|---|
| | FIA_UIA_EXT.1 | User Identification and Authentication |
| Security Management | 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.

The TOE provides auditing of all administrator actions and of all events explicitly listed in Table 12 that occur within the CLI and Management Console 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 12**.**

The TOE supports the SSH, TLS, and HTTPS protocols and will record 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 ten (10) megabytes of data before it will begin to overwrite the earliest audited events. A Privileged Administrator can modify the maximum local audit log storage to suit the deployment. The TOE provides the ability to securely transmit audit logs to an external audit server using TLS. The audit server, (installed with an HTTP command line tool such as Wget or cURL) uses a script to periodically issue the following command to retrieve audit logs: `https://<TOE_Address>:8082/Eventlog/fetch=0xffffffff`. To retrieve the audit logs, this command must be accompanied with the credentials of an Authorized Administrator or the retrieval will fail. A copy of the entire Audit Log is sent encrypted using TLS to the audit server where the Audit Logs contents can be verified and viewed. Retrieving a copy of the audit logs does not delete the audit logs stored within the TOE. All encryption is provided by CAVP-validated algorithms. Only authenticated administrator roles have access to the audit logs. No users, including authenticated administrators, have the ability to modify or delete the audit logs within the TOE. 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, and key generation are provided by the CAVP validated Cavium CN501 (AES Cert. #105), CN1610 (AES Cert. #1265, HMAC Cert. #736), and CN1620 (AES Cert. #1265, HMAC Cert. #736) HACs, and Blue Coat's proprietary CAVP validated cryptographic algorithms (AES Cert. #2925, RSA Cert. #1534, SHA Cert. #2461, DRBG Cert. #539, and HMAC Cert. #1852). The TOE uses SSH, TLS, and HTTPS (via TLS) to protect communications. SSH provides a trusted path for remote administrators accessing the TOE's CLI. TLS is used to provide a trusted channel for ProxySG requests to a Lightweight Directory Access Protocol (LDAP) server and to BCAAA. TLS is also used to provide a trusted channel during audit log transmissions from the TOE. HTTPS (via TLS) is used to provide a trusted path for administrator management connections to the TOE's Management Console. The TOE uses symmetric AES keys to encrypt and decrypt data. The TOE also provides HMAC[38]-SHA[39] and SHS[40] to support TOE cryptographic functionality.

The CAVP validated Cavium CN501 (AES Cert. #105), CN1610 (AES Cert. #1265), and CN1620 (AES Cert. #1265) HACs are capable of performing encryption and decryption using the AES-128-CBC and AES-256-CBC algorithms. The CAVP validated Cavium CN1610 (, HMAC Cert. #736) and CN1620 (, HMAC Cert. #736) are capable of performing Message Authentication Code operations using the HMAC-SHA-512 algorithm. The TOE's power-up cryptographic self-tests are also implemented on all the Cavium HACs. The AES known answer test (KAT) is first tested on the HAC, and if the implementation passes the test, all cryptographic operations involving AES-128-CBC or AES-256-CBC will be performed on the HACs. If the AES KAT fails for the Cavium HAC, the HAC is disabled and AES operations take place exclusively in the software (provided the software implementation self-tests all successfully pass). This same process repeats for the HMAC KAT; however, HMAC is not implemented on the CN501 Cavium HAC and therefore this KAT will occur only on the CN1610 and CN1620 for the ProxySG 900 and ProxySG 9000 respectively. For a complete list and description of the self-tests performed by the TOE, please see Table 15 below.

[38] HMAC – (keyed-) Hashed Message Authentication Code
[39] SHA – Secure Hash Algorithm
[40] SHS – Secure Hash Standard

The TOE's cryptographic module is capable of generating cryptographic keys that provide at least 112 bits of symmetric key strength, in accordance with FIPS standards. Keys are generated via the use of a CTR_DRBG[41] (using AES-256) to provide random keying material. The TOE implements RSA key pair establishment conformant to section 6.2 of NIST Special Publication 800-56B. The TOE implements RSA key pair generation that provide at least 112 bits of strength conformant to section 6.3 of NIST Special Publication 800-56B.

The TOE generates a default key ring (containing a public/private RSA 2048-bit key pair and a certificate or certificate signing request) when the TOE boots from the uninitialized state. The key ring is used to secure SSH, TLS, and HTTPS sessions with the Management Console.

The MAK is generated internally using the CAVP validated SP 800-90A CTR_DRBG and stored in the TOE's EEPROM in plaintext. The MAK is an AES CBC[42] 256-bit key that never exits the module and is overwritten with zeros when the TOE is configured to stop using CAVP validated cryptography. It is used to encrypt the TOE's private RSA key, and local authentication passwords.

The TOE can use AES 128, 192, 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. HTTPS/TLS is used for management sessions via the Management Console. TLS is used to protect communications with a remote authentication and the management workstation acting as an audit server.

The TOE supports the following mandatory TLS ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA

The TOE also supports the following optional TLS ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256

The TOE's cryptographic module is also used when the TOE has been configured to require X.509 certificate-based authentication, such as for environments where CAC[43] authentication is used. Under these circumstances the TOE will implement specially configured CPL during administrator authentication in order to facilitate TLS mutual authentication. This is accomplished by modifying the HTTPS-Console service so that it can be configured to validate a client certificate against a chosen certificate authority (CA) list. X.509 certificate-based authentication will take place against a Certificate realm, and administrator authorization takes place against a local or configured LDAP realm.

The X.509 certificate-based authentication process is as follows:

1. The administrator opens a browser and establishes a clear-text HTTP connection with the TOE.
2. Using CPL similar to the VPM `NotifyUser` action, the administrator is presented with a DoD warning banner which they must positively acknowledge and accept.
3. `NotifyUser` redirects the administrator's browser to an HTTPS connection with the TOE that requires mutual authentication. This is made possible by CPL that puts the TOE in reverse-proxy mode at this point.

[41] DRBG –Deterministic Random Bit Generator
[42] CBC – Cipher Block Chaining
[43] CAC Authentication was not tested; only X.509 certificate-based authentication was tested in the CC Evaluated Configuration.

4. The TLS handshakes begin. The reverse-proxy service on the TOE requires a certificate to complete the handshake (i.e. the `verify-peer` setting has been enabled in the reverse-proxy service).
5. The administrator's browser presents the administrator with a dialog box prompting the administrator to select a certificate.
6. The administrator selects the X.509 certificate.
7. The TOE authenticates the X.509 certificate against the CA list that has been configured on the reverse proxy service using local CRLs and OCSP to check for certificate revocation.
8. The administrator reviews and accepts the certificate issued to the web browser by the TOE. A mutually authenticated TLS session is now in use.
9. The TOE extracts the user's subject name from the `subjectAltNames` extension of the X.509 certificate according to configuration of the certificate realms. Within the `subjectAltNames` extension is the user's `userPrincipleName` (UPN) or CommonName (CN) field. The UPN/CN is extracted from the certificate and is checked against the Principle Name (PN) field of a user record in Active Directory (AD), the LDAP server.
10. The certificate realm is configured to use an LDAP realm for authorization. The LDAP user is determined by LDAP search using the following filter: `(userPrincipleName=$(user.name)`).

The administrator is granted access to the Management Console if the UPN/CN is found in the LDAP directory. The exchanges with the LDAP server are secured using TLS. Conditions like *group=* and *ldap.attribute <name>* may also be used to authorize the user and to specify if the user should have read-only or read-write access.

The TOE uses the open source OpenSSH implementation of the SSHv2 protocol which conforms to RFCs 4251, 4252, 4253, and 4254 as shown here: http://www.openssh.org/specs.html. The TOE supports the use of the RSA public key algorithm (SSH_RSA) and password-based mechanisms 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 kilobytes, 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 and HMAC-SHA1-96 are available. Diffie-Hellman-group14-SHA1 is the only allowed key exchange method used for the SSH protocol.

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 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. The private RSA key and local authentication passwords are all persistent while the module is operating in the evaluated configuration. These CSPs (which are stored encrypted with the MAK) can be zeroized by disabling the TOE's use of CAVP validated cryptography. Additionally, when the TOE is configured to discontinue use of the CAVP validated cryptography, the memory location in EEPROM where the MAK resides is overwritten with zeros, effectively making any keys or passwords encrypted with the MAK permanently inaccessible.

**TOE Security Functional Requirements Satisfied:** FCS_CKM.1, FCS_CKM_EXT.4, FCS_DRBG_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 only the size of incoming packet's buffer area is used for the outgoing packet's buffer area, ensuring any residual data in the buffer area used by previous network packets is made unavailable (through overwriting or exclusion) during buffer allocation for the outgoing packet. When a network packet is received by the TOE, the TOE's Network Interface Card (NIC) writes the packet's contents into memory buffers that are used for network packet processing. The contents of the memory buffers will be overwritten with the contents of the received packet, ensuring any user data that was previously present, is no longer available in the memory buffer for intentional or unintentional reuse. During the allocation of the memory buffer, because the memory buffers may be larger than received packet, the TOE uses the incoming packet size to track what it considers 'good' data. If a larger packet is received followed by a smaller packet, the TOE will update what it considers 'good' data to match the size of the received smaller packet. When the packet is sent out, the NIC reads directly from the memory buffer and only reads up to the 'good' data size ensuring that any data from the previous larger packet will not find its way into a new packet. This guarantees that there is no possibility for residual data from the memory buffer's previous contents to find its way into a new packet.

**TOE Security Functional Requirements Satisfied:** FDP_RIP.2.

### 7.1.4 Identification and Authentication

The TOE provides mechanisms for authenticating administrators connecting to the TOE through the CLI and Management Console. When an administrator connects through the Management Console, the TOE must authenticate them. The TOE will consult the internal authentication mechanism in place to authenticate the administrator. The TOE can authenticate administrators through the Management Console using a Certificate realm (which uses locally configured X.509 certificates for authentication, and also using LDAP over TLS), an IWA-BCAAA realm (which uses BCAAA for authentication against a remote Windows Active Directory), an LDAP realm (which uses an external LDAP server for authentication), and the Local realm (which uses locally stored username and password combinations for authentication). The TOE can locally authenticate administrators through the CLI over SSH using a username and password combination (via the predefined 'admin' account or the Local realm), the IWA-BCAAA realm, the LDAP realm, and a localized RSA keypair. The TOE can locally authenticate administrators through the serial console using the predefined 'admin' account only.

The authentication mechanism used by the TOE for administrator authentication can only be modified by another administrator through CPL. Using CPL, an authorised administrator can craft policies controlling administrative access by users (excluding administrators authenticating with the 'admin' account credentials, which are not subject to crafted CPL). This allows administrative access to be granted or denied based on the username, the groups to which the user belongs, and the time of day.

If authentication via Certificate realm or IWA-BCAAA realm fails, or if CPL has not been configured to perform such an authentication for administrators, then the TOE will prompt for a username and a password to be used against the Local realm.

The TOE can be configured to support X.509 certificate-based authentication; this type of configuration would be required if the TOE was deployed in a CAC enabled environment; however CAC was not tested. Please refer to section 7.1.2 for a detailed description on X.509 certificate-based authentication.

Administrator authentication is enforced through the use of a password, X.509 certificate, or an RSA key. Authorized administrators 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 ( )*. All accounts on the TOE, including the 'admin' account, are subject to the configured password policy.

All forms of authentication for the CLI and Management Console are secured using a trusted path or trusted channel depending on the authentication mechanism in use. The CLI only accepts credentials via a serial connection or an SSH session. The Management Console interface only accepts credentials via HTTPS (over TLS). When administrator authentication is configured to use an IWA-BCAAA Realm or an LDAP realm that is external to the TOE, the connection used to transmit the authentication credentials is secured using TLS to the external authentication server (the LDAP server or Microsoft AD Domain Controller where the BCAAA is installed).

A login is considered successful if the credentials submitted by the administrator can be validated by the TOE. If authentication using username and password credentials is used, and the credentials match a locally stored username and password, or authentication via the IWA-BCAAA realm or the LDAP realm succeeds, login is considered successful. If the Certificate realm is used for authentication, the TOE must first verify the submitted X.509 certificate is signed by an explicitly trusted Certificate Authority and that the client possesses the private key that corresponds with the public key in the submitted X.509 certificate. The TOE will encrypt a message using the public key and send it to the client attempting authentication. The client will use its private key to decrypt the TOE's encrypted message. The TOE receives the client's response which should contain the decrypted message. If the TOE is able to verify the decrypted message, login is considered successful. Administrators are notified by the CLI and Management Console when there is a failure in authentication and they will be prompted to try again.

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

### 7.1.5 Security Management

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

The TOE defines two Administrator roles:

1. *Standard* or *Unprivileged* mode Administrator – has not been granted access to the "enabled" mode in the CLI and has been given "read-only" privileges when using the Management Console. The *Standard* or *Unprivileged* mode Administrator will access the CLI and Management Console interfaces for management of the module; however, the Administrator cannot make any changes to configuration settings. When the *Standard* or *Unprivileged* mode Administrator is administering the module over the Management Console, they perform all the same services available in CLI ("standard" mode only services) and additionally, can query the TOE to determine if CAVP validated cryptography is in use in the Management Console only.

2. *Enabled,* or *Privileged* mode Administrator – has been granted "enabled" mode access while using the CLI and "read/write" access while using the Management Console. When the *Enabled,* or *Privileged* mode Administrator is using the CLI, and while in the "enabled" mode of operation, *Enabled,* or *Privileged* mode Administrators may configure the TOE's use of CAVP validated cryptography (local serial port only) and query if the TOE is configured to CAVP validated cryptography. In addition, this role may do all the services available to *Standard* Administrators while not in "enabled" mode. Once the Administrator has entered the "enabled" mode, the *Enabled,* or *Privileged* mode Administrator may then enter the *Configuration* mode via the CLI. The *configuration* mode provides the Administrator management capabilities to perform tasks such as account Management Console, they can perform all the same services available in CLI (equivalent to being in the *Configuration* mode in the CLI) except the *Enabled,* or *privileged* mode Administrator is unable to configure the TOE to use CAVP validated cryptography.

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.

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

### 7.1.6 Protection of the TSF

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

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. 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 SGOS 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.

The TOE generates its own time stamps that originate from a system hardware clock. The time stamps are considered reliable because the TOE has successfully completed its bootup and initialization process, indicating that all hardware is functioning properly. Had there be a failure with the hardware clock, the TOE would not have completed its bootup cycle. Additionally, once operational, the time stamps are reliable because only an Authorized Administrator is capable of modifying the clock. Because the Authorized Administrator is trusted to follow and apply all administrator guidance while administering the TOE, the time stamps are reliable and will not be maliciously changed.

The timestamp is used by the audit logs to record an accurate time for each auditable event and must be set to the current Coordinated Universal Time (UTC). The clock can be changed through the CLI and Management Console. Using the Management Console, an authorized Administrator may edit the time by navigating to the **Configuration > General > Clock >Clock tab** page. Only Authorized Administrators may edit the time and the value of the timestamps is reliable. Use of an NTP server is not part of the evaluated configuration. Administrators using the CLI may also edit the time by entering the *enabled* mode, followed by the *configuration* mode and using the "`clock`" command and correct parameters.

Administrators can find the current version of TOE software by going to the home page of the Management Console or using the `show version` command through the CLI. 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 public key) 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.

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. The TOE contains both a software implementation of its cryptographic algorithms and a HAC that contains a subset (only AES and HMAC) of the cryptographic algorithms. The TOE first verifies correct operation of the software implementation, and then tests the HAC for its cryptographic functionality.

The TOE proceeds to test its software implementation of cryptographic functionality (using the tests in Table 15 below) 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. After successful completion of the software self-tests, the HAC self-tests are executed (using the tests in Table 15 below marked with 'HAC'). If the HAC self-tests pass, the TOE will use the HAC for cryptographic services; if the HAC self-tests fail, a message stating "Hardware Crypto Accelerator failed FIPS self-tests: disabled" is logged to the audit log and the HAC is disabled. If the HAC is disabled, the TOE continues to operate securely and will use the software implementation exclusively. Therefore, successfully passing the startup self-tests is sufficient to show that the TSF is operating correctly and that no errors resulted from the tests.

A description of each self-test is given in Table 15 below.

**Table 15 Self-Test Descriptions**

| Self-Test | Description |
|---|---|
| AES KAT(software) | The KAT encrypts a known plaintext with known keys. It then compares the resultant ciphertext with the expected ciphertext hard-coded in the TOE. If the two values differ, then the KAT fails. If the two values agree, the AES KAT then decrypts the ciphertext with the known keys and compares the decrypted text with the known plaintext. If they differ, then the test fails. If they are the same, then the test passes. |
| RSA Digital Signature Generation and Verification KAT | The private key is used to sign a block of data, and the resultant value is compared with the original data. If they are the same, the test fails. If they differ, then the public key is used to verify the ciphertext, and the output is compared to the original data. If they are the same, the test passes. Otherwise, it is failed. |
| RSA Pair-wise Consistency Test | The RSA pair-wise consistency test for key wrapping uses an RSA private key to wrap the hash of some data. The resulting wrapped data is compared to the original hashed data before it was wrapped. If the two values are equal, then the test has failed. If the two values differ, the public key is used to unwrap the hashed data and the resulting value is compared to the original hashed data. If the two values are not equal, the test has failed. |
| SHA-1 KAT | The KAT takes a specific value and hashes it. This digest value is then compared to the known value. If the values differ, the test fails. If they are the same, the test passes. |
| SHA-224 KAT | The KAT takes a specific value and hashes it. This digest value is then compared to the known value. If the values differ, the test fails. If they are the same, the test passes. |
| SHA-256 KAT | The KAT takes a specific value and hashes it. This digest value is then compared to the known value. If the values differ, the test fails. If they are the same, the test passes. |

| Self-Test | Description |
|---|---|
| SHA-384 KAT | The KAT takes a specific value and hashes it. This digest value is then compared to the known value. If the values differ, the test fails. If they are the same, the test passes. |
| SHA-512 KAT | The KAT takes a specific value and hashes it. This digest value is then compared to the known value. If the values differ, the test fails. If they are the same, the test passes. |
| HMAC SHA-1 KAT (software) | The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes. |
| HMAC SHA-224 KAT (software) | The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes. |
| HMAC SHA-256 KAT (software) | The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes. |
| HMAC SHA-384 KAT (software) | The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes. |
| HMAC SHA-512 KAT (software) | The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes. |
| CTR_DRBG Self-Test | A known seed value is used to initialize the DRBG. A block of random data is then generated and compared to a pre-generated value. If these values are the same, the test is passed. Otherwise, the test is failed. |
| AES KAT (HAC)<br><br>Present on all TOE hardware models | The KAT encrypts a known plaintext with known keys. It then compares the resultant ciphertext with the expected ciphertext hard-coded in the TOE. If the two values differ, then the KAT fails. If the two values agree, the AES KAT then decrypts the ciphertext with the known keys and compares the decrypted text with the known plaintext. If they differ, then the test fails. If they are the same, then the test passes.<br><br>If this test passes, AES operations will be performed by the Cavium HACs. If this test fails, the software implementation will be relied upon to provide AES encryption and decryption operations. |
| HMAC SHA-1 KAT (HAC)<br><br>Present only on the ProxySG SG900 and SG9000 | The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes. |
| HMAC SHA-224 KAT (HAC)<br><br>Present only on the ProxySG SG900 and SG9000 | The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes. |
| HMAC SHA-256 KAT | The KAT creates a MAC using a known message and known key. This MAC |

| Self-Test | Description |
|---|---|
| (HAC)<br><br>Present only on the ProxySG SG900 and SG9000 | value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes. |
| HMAC SHA-384 KAT (HAC)<br><br>Present only on the ProxySG SG900 and SG9000 | The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes. |
| HMAC SHA-512 KAT (HAC)<br><br>Present only on the ProxySG SG900 and SG9000 | The KAT creates a MAC using a known message and known key. This MAC value is then compared to the expected MAC value. If the values differ, the test fails. If they are the same, the test passes. |

**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. Local sessions must be initiated by accessing the CLI via the serial port. Remote sessions may be initiated by accessing the CLI using SSH or accessing the Management Console using HTTPS via TLS. Administrators may also terminate their sessions voluntarily. 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 Management Console and the CLI.

**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 CLI over SSH and the Management Console over 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.

Additionally, the TOE provides a trusted channel between the TOE and the trusted IT entities used for the audit and authentication servers. The TOE protects audit log traffic by encrypting it with a secure TLS tunnel. For authentication mechanisms that require the use of LDAP or the BCAAA, the communication between the TOE and the authentication server is also protected with TLS. The TLS channel prevents unauthorized disclosure and detection of modification for all audit and authentication data sent to the Administrator's management workstation and authentication server respectively.

The TOE does not communicate with any other servers or network devices in the evaluated configuration.

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

# 8 Rationale

## 8.1 Conformance Claims Rationale

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

### 8.1.1 Variance Between the PP and this ST

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

- An Application Note in the NDPP states that the word "manage" in FMT_MTD.1 is the default requirement for management of TSF data. Other iterations are possible. A table was added to FMT_MTD.1 to include the operations listed in the Application Note for manage and any other operations administrators in the TOE can perform.

### 8.1.2 Security Assurance Requirements Rationale

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

### 8.1.3 Dependency Rationale

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

**Table 16 Functional Requirements Dependencies**

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

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

| SFR ID | Dependencies | Dependency Met | Rationale |
|---|---|---|---|
| FMT_SMF.1 | FPT_TUD_EXT.1 | ✓ | |
| | FCS_COP.1(2) | ✓ | |
| | FIA_UIA_EXT.1 | ✓ | |
| 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 | FIA_UAU.1 | ✓ | Although FIA_UAU.1 is not included, FIA_UIA_EXT.1 provides coverage for user identification and authentication which supersedes FIA_UAU.1. |
| FTA_TAB.1 | No dependencies | ✓ | |
| FTP_ITC.1 | FCS_HTTPS_EXT.1 | ✓ | |
| | FCS_TLS_EXT.1 | ✓ | |
| FTP_TRP.1 | FCS_SSH_EXT.1 | ✓ | |
| | FCS_TLS_EXT.1 | ✓ | |
| | FCS_HTTPS_EXT.1 | ✓ | |

# 9 Acronyms and Terms

This section describes the acronyms and terms.

## 9.1 Terminology

**Table 17 Terms**

| Name | Definition |
|---|---|
| **Authorized Administrator** | A user with administrator TOE access that has been successfully identified and authenticated by the TOE. Can be either a *Standard* or *Privileged* mode Administrator. |
| **Target network** | The domain of network and managed devices to be analyzed by the TOE. |

## 9.2 Acronyms

**Table 18 Acronyms**

| Acronym | Definition |
|---|---|
| ADN | Application Delivery Network |
| AES | Advanced Encryption Standard |
| ANSI | American National Standards Institute |
| AOL | America Online |
| BCAAA | Blue Coat System Authentication and Authorization Agent |
| CA | Certificate Authority |
| CAC | Common Access Card |
| CBC | Cipher Block Chaining |
| CC | Common Criteria |
| CEM | Common Criteria Evaluation Methodology |
| CFB | Cipher Feedback |
| CIFS | Common Internet File System |
| CLI | Command Line Interface |
| CM | Configuration Management |
| CPL | Content Policy Language |
| CRL | Certificate Revocation List |
| CTR | Counter mode |
| DNS | Domain Name System |
| DoD | United States Department of Defence |
| DRBG | Deterministic Random Bit Generator |
| DSA | Digital Signature Algorithm |
| EAL | Evaluation Assurance Level |
| ECB | Electronic Codebook |
| FIPS | Federal Information Processing Standard |
| FSP | Functional Specification |
| FTP | File Transfer Protocol |
| GB | Gigabyte |
| GigE | Gigabit Ethernet |
| HMAC | (keyed) Hashed Message Authentication Code |
| HSPD | Homeland Security Presidential Directive |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |

| Acronym | Definition |
| --- | --- |
| HTTPS | Secure Hypertext Transfer Protocol |
| ICC | Integrated Circuit Card |
| IMAP | Internet Message Access Protocol |
| IP | Internet Protocol |
| IT | Information Technology |
| IWA | Integrated Windows Authentication |
| KAT | Known Answer Test |
| LAN | Local Area Network |
| LDAP | Lightweight Directory Access Protocol |
| MAC | Message Authentication Code |
| MAK | Master Appliance Key |
| MAPI | Messaging Application Programming Interface |
| MIME | Multipurpose Internet Mail Extensions |
| MMS | Microsoft Media Streaming |
| MSN | The Microsoft Network |
| NCSA | National Center for Supercomputing Applications |
| NDPP | Security Requirements for Network Devices v1.1 Protection Profile |
| NIC | Network Interface Card |
| NTP | Network Time Protocol |
| OCS | Original Content Server |
| OFB | Output Feedback |
| OS | Operating System |
| OSP | Organizational Security Policy |
| PCI-e | Peripheral Controller Interconnect –Express |
| PIN | Personal Identification Number |
| POP3 | Post Office Protocol version 3 |
| PP | Protection Profile |
| RBG | Random Bit Generation |
| RFC | Request For Comments |
| RNG | Random Number Generator |
| RSA | Rivest, Shamir, and Adelman |
| RTSP | Real-Time Streaming Protocol |
| SAR | Security Assurance Requirement |
| SAS | Serial Attached SCSI |

| Acronym | Definition |
|---|---|
| SATA | Serial Advanced Technology Attachment |
| SCSI | Small Computer System Interface |
| SFP | Security Functional Policy |
| SFR | Security Functional Requirement |
| SGOS | Secure Gateway Operating System |
| SHA | Secure Hash Algorithm |
| SHS | Secure Hash Standard |
| SMTP | Simple Mail Transfer Protocol |
| SOCKS | SOCKet Secure |
| SP | Special Publication |
| SSH | Secure Shell |
| SSL | Secure Sockets Layer |
| ST | Security Target |
| TCP | Transmission Control Protocol |
| TB | Terabyte |
| TLS | Transport Layer Security |
| TOE | Target of Evaluation |
| TSF | TOE Security Function |
| TSFI | TOE Security Functional Interface |
| U | Unit |
| UTC | Coordinated Universal Time |
| URL | Uniform Resource Locator |
| VPM | Visual Policy Manager |
| WAN | Wide Area Network |

Prepared by:
**Corsec Security, Inc.**

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

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