Exabeam Security Management Platform

Security Target

ST Version: 1.0
July 26, 2019

Prepared By:

Booz | Allen | Hamilton

delivering results that endure

Exabeam, Inc.
2 Waters Park Dr., Suite 200
San Mateo, CA 94403

Cyber Assurance Testing Laboratory
1100 West St.
Laurel, MD 20707

Table of Contents

Table of Tables

1 Security Target Introduction

This chapter presents the Security Target (ST) identification information and an overview. An ST contains the Information Technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by the TOE.

1.1 ST Reference

This section provides information needed to identify and control this ST and its Target of Evaluation.

1.1.1 ST Identification

ST Title:	Exabeam Security Management Platform Security Target
ST Version:	1.0
ST Publication Date:	July 26, 2019
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this document provides identifying information for the ST and TOE as well as a brief description of the TOE and its associated TOE type.

Chapter 2 describes the TOE in terms of its physical boundary, logical boundary, exclusions, and dependent Operational Environment components.

Chapter 3 describes the conformance claims made by this ST.

Chapter 4 describes the threats, assumptions, objectives, and organizational security policies that apply to the TOE.

Chapter 5 defines extended Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs).

Chapter 6 describes the SFRs that are to be implemented by the TSF.

Chapter 7 describes the SARs that will be used to evaluate the TOE.

Chapter 8 provides the TOE Summary Specification, which describes how the SFRs that are defined for the TOE are implemented by the TSF.

1.1.3 Terminology

This section defines the terminology used throughout this ST. The terminology used throughout this ST is defined in Table 1. These tables are to be used by the reader as a quick reference guide for terminology definitions.

Term	Definition
Security Administrator	The claimed Protection Profile defines a Security Administrator role that is authorized to manage the TOE and its data. For the TOE, this is considered to be the Exabeam user for the local or remote CLI, the root user for the local CLI, and any user with the permissions provided to the 'Administrator' role for the GUI.
Trusted Channel	An encrypted connection between the TOE and a system in the Operational Environment.
Trusted Path	An encrypted connection between the TOE and the application an Authorized Administrator uses to manage it (web browser, terminal client, etc.).
User	In a CC context, any individual who has the ability to manage TOE functions or data.

Table 1: CC Specific Terminology

1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 2. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
CA	Certificate Authority
CC	Common Criteria
CLI	Command-line Interface
CN	Common Name
CRNGT	Continuous Random Number Generator Test
CPU	Central Processing Unit
CVL	Component Validation List
DSS	Digital Signature Standard
DN	Distinguished Name
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
KAS	Key Agreement Scheme
KAT	Known Answer Test
KDF	Key Derivation Function
NDcPP	collaborative Protection Profile for Network Devices, version 2.0 + Errata 20180314
NIAP	National Information Assurance Partnership
OCSP	Online Certificate Status Protocol
OS	Operating System
POST	Power-On Self Test
PP	Protection Profile
PKCS	Public-Key Cryptography Standards
RBAC	Role Based Access Control

SAN	Subject Alternative Name
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFTP	Secure File Transfer Protocol
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SMP	Security Management Platform
SSL	Secure Sockets Layer
SSH	Secure Shell
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function

Table 2: Acronyms

1.1.5 Reference

[1] collaborative Protection Profile for Network Devices, version 2.0 + Errata 20180314 (NDcPP)

[2] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-001

[3] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-002

[4] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-003

[5] Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-004

[6] NIST Special Publication 800-38A Recommendation for Block Cipher Modes of Operation, December 2001

[7] FIPS PUB 140-2 Federal Information Processing Standards Publication Security Requirements for Cryptographic Modules May 25, 2001

[8] FIPS PUB 180-4 Federal Information Processing Standards Publication Secure Hash Standard (SHS) August 2015

[9] FIPS PUB 186-4 Federal Information Processing Standards Publication Digital Signature Standard July 2013

[10] FIPS PUB 197 Advanced Encryption Standard November 26 2001

[11] FIPS PUB 198-1 Federal Information Processing Standards Publication The Keyed-Hash Message Authentication Code (HMAC) July 2008

1.2 TOE Reference

The Target of Evaluation (TOE) is the Exabeam Security Management Platform (SMP). Exabeam's SMP contains two models that communicate with each other in the evaluated configuration making it a distributed TOE. The first model is the EX3000 which has the Data Lake software installed. The second model is the EX4000 which has the Advanced Analytics software and Incident Responder software

installed. In the evaluated configuration, there is only a single model of the EX3000 and a single model of the EX4000 within the distributed TOE.

1.3 TOE Overview

The TOE is the Exabeam Security Management Platform (SMP) product referred to as just SMP or TOE from this point forward. The TOE allows Security Administrators access through a local CLI (keyboard/monitor), remote CLI via SSH, and a GUI via TLS/HTTPS. The TOE was evaluated against the Security Functional Requirements defined in Section 6.3 only.

Exabeam SMP's primary functionality is to collect network traffic and events, correlate the data collected to detect threats, and provide recommendations for responses to safeguard the network against cyberattacks. The SMP model with the Data Lake software provides the capability to collect the network traffic and events and will send that data to the other TOE component over TLS for threat detection and response recommendation. The SMP model receiving the collected events has the Advanced Analytics software which will detect threats and the Incident Responder software that will create response actions that the network administrator can perform to mitigate the threat.

The TOE was evaluated as a network device only and SMP's network event collection, threat detection, and incident response functionality above were not assessed during this evaluation.

The following figure depicts the TOE boundary:

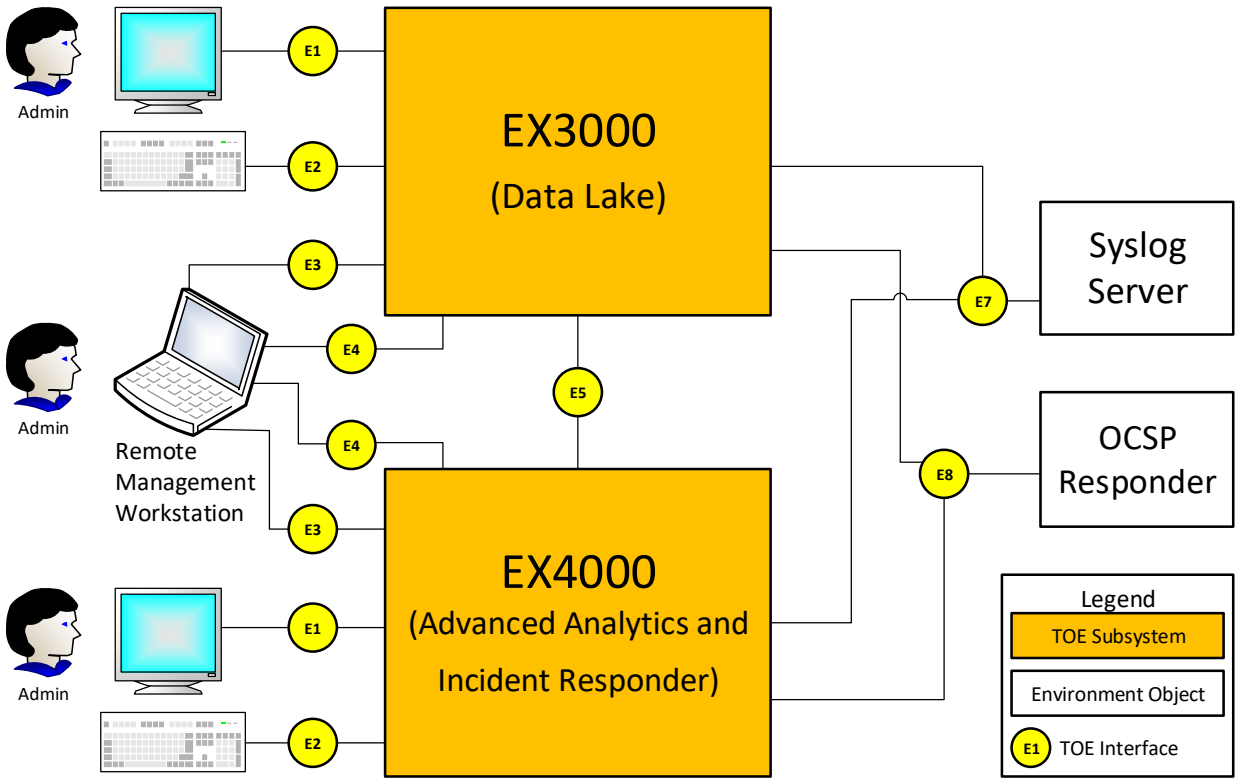

Figure 1-1: TOE Boundary

As illustrated in Figure 1, Exabeam's SMP provides Security Administrators the ability to manage the TOE both locally and remotely. Each TOE model has connections for a monitor (E1) and keyboard (E2) for local CLI management. Remote administration is accomplished by either using an SSH client to connect to the remote CLI (E3) or using a web browser to connect to the GUI (E4) which is protected by TLS/HTTPS. The TOE also connects to multiple servers in its Operational Environment which support its normal functions. The TOE transfers audit records to a remote syslog server (E7) via TLS. An Online Certificate Status Protocol (OCSP) Responder (E8) is used to determine the validity of certificates provided by an entity in the Operational Environment when connecting to the TOE. The only internal connection for the distributed TOE is a TLS connection between the EX3000 and EX4000 models (E5) for the sending collected network event data from EX3000 to the EX4000.

1.4 TOE Type

The TOE is a network device and is as defined in the NDcPP which states: "This is a Collaborative Protection Profile (cPP) whose Target of Evaluation (TOE) is a network device… A network device in the context of this cPP is a device composed of both hardware and software that is connected to the network and has an infrastructure role within the network." The TOE consists of the Exabeam SMP model EX3000 with Data Lake software and EX4000 model with Advanced Analytics software and Incident Responder software. Thus, the TOE is a network device composed of hardware and software. Within the infrastructure of the network, SMP collects network traffic and events, correlates the data collected to detect threats, and provides recommendations for responses to safeguard the network against cyber attacks. Because the device is connected to and has an infrastructure purpose within the network, this conformance claim is appropriate. Additionally, since SMP contains multiple components communicating with each other, it is a distributed TOE based upon the requirements of the NDcPP.

2 TOE Description

This section provides a description of the TOE in its evaluated configuration. This includes the physical and logical boundaries of the TOE.

2.1 Evaluated Components of the TOE

The TOE is the Security Management Platform (SMP). The following table describes the TOE components in the evaluated configuration:

Component	Definition
EX3000	Model with the Data Lake software installed
EX4000	Model with the Advanced Analytics and Incident Responder software installed

Table 3: Evaluated Components of the TOE

2.2 Components and Applications in the Operational Environment

These components and the functionality they provide are outside the scope of evaluation testing but are needed to support the tested functionality of the TOE. The following table lists components and applications that are used in the operational environment for the TOE's evaluated configuration.

OE Component	Definition
Management Workstation	Any general-purpose computer that is used by an administrator to manage the TOE. For the TOE to be managed remotely the management workstation is required to have: • Browser to access the TOE's GUI • SSHv2 client to access the TOE's secure shell command-line interface The TOE's secure shell command line interface can also be accessed locally with a physical connection to the TOE using a keyboard and monitor.
Syslog Server	The TOE connects to a syslog server to send syslog messages for remote storage via TLS connection where the TOE is the TLS client. This is used to send copies of audit data to be stored in a remote location for data redundancy purposes.
OCSP Responder	A server deployed within the Operational Environment which confirms the validity and revocation status of certificates.

Table 4: Supporting Components in the Operational Environment

2.3 Excluded from the TOE

The following TOE functionality, components, and/or applications are not included in the evaluated configuration. They provide no added security related functionality for the evaluated product. They are separated into three categories: not installed, installed but requires a separate license, and installed but not part of the TSF.

2.3.1 Not Installed

This section contains components or software that were not installed for this evaluation:

- Collectors – software that collects data from various sources

2.3.2 Installed but Requires a Separate License

There are no excluded components, applications, and or functionality that are installed and require a separate license for activation.

2.3.3 Installed But Not Part of the TSF

This section contains functionality that is part of the purchased product but is not part of the TSF relevant functionality that is being evaluated as the TOE based on the Protection Profile.

- Ability to collect network traffic and events, correlate the data collected to detect threats, and provide recommendations for responses to the attacks

2.4 Physical Boundary

2.4.1 Hardware

The physical boundary of the TOE is defined in the following table:

Model Number	EX3000	EX4000
Size	1 RU	1 RU

Power	AC	AC
Processor	Intel Xeon E5-2620	Intel Xeon E5-2690
Memory (RAM)	192GB DDR4 2666MHz (6 x 32GB)	256GB DDR4 2400MHz (8 x 32GB)
Storage	• 9x Seagate EC3.5v5 4TB SATA 512E 6Gbps SATA3 7200rpm 128MB 3.5i • 2x Samsung PM863a 1.92TB SSD • 1x Intel S4500 240GB SSD • Maximum Storage Capacity: 35.6TiB • Maximum Usable Capacity: 27.5TiB	• 1x Intel S3500 150GB SSD • 3x Samsung PM863A 960GB SSD • 6x Seagate EC2.5 2TB HDD

Table 5: Hardware

2.4.2 Software

The TOE is the Exabeam Security Management Platform. The TOE's software version is Core (PLT-i10) which includes the Data Lake (EX3000), and Advanced Analytics and Incident Responder (EX4000) software. The underlying software of the TOE runs on CentOS 7.6 and includes the OpenSSL 6.0 cryptographic module.

2.5 Logical Boundary

The TOE is comprised of the following security features that have been scoped by the NDcPP.

1. Security Audit
2. Cryptographic Support
3. Communication
4. Identification and Authentication
5. Security Management
6. Protection of the TSF
7. TOE Access
8. Trusted Path/Channels

2.5.1 Security Audit

Audit records are generated on each model for various types of management activities and events that occur on that model. These records include the date and time stamp of the event, the event type, and the subject identity. Audit records are stored in rsysreceived.log on each TOE model and can be configured to also be sent to a syslog server via a TLS connection. When the storage space allocated to rsysreceived.log

is exhausted, the model will delete the oldest log file, archive the previous active file, and generate a new active file to which audit records are written.

2.5.2 Cryptographic Support

Each TOE model provides cryptography in support of communications between itself and the Operational Environment. The protocols used for this are TLS, HTTPS, and SSH. The TOE uses TLS to secure the automatic transfer of syslog audit records. TLS/HTTPS is used to secure the connection for remote management of the TOE via the GUI and SSH is used to secure the remote CLI interface for remote management of the TOE. TLS mutual authentication is used for communication between TOE components.

Exabeam's implementation of these has been validated to ensure that the algorithms are appropriately strong for use in trusted communications. The TOE collects entropy from sources contained within the device to ensure sufficient randomness for secure key generation.

Cryptographic keys are generated using the CTR_DRBG provided through this module and the references to the keys are destroyed when no longer needed.

The following table lists the CAVP algorithm certificates for the OpenSSL 6.0 cryptographic module:

SFR	Algorithm/Protocol	OpenSSL CAVP Cert #
FCS_CKM.1	RSA FIPS 186-4 Key Generation	2786
FCS_CKM.1	DSA FIPS 186-4 Key Generation for Diffie-Hellman FFC	1346
FCS_CKM.2	FFC Key Establishment 2048 bits (CVL)	1687
FCS_COP.1/ DataEncryption	AES-128-CBC, AES-256-CBC, AES-256-GCM, AES-256-CTR	5203
FCS_COP.1/ SigGen	RSA FIPS 186-4 Signature Generation and Signature Verification	2786
FCS_COP.1/Hash	SHA-1, SHA-256, SHA-384, SHA-512	4193
FCS_COP.1/KeyedHash	HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	3445
FCS_RBG_EXT.1	CTR_DRBG (AES)	1975

Table 6:Cryptographic Algorithm Table for OpenSSL

2.5.3 Communication

In order for the EX3000 to send collected network events to the EX4000, the Security Administrator must have configured these two components to communicate. The Security Administrator also has the ability to disable communication between the TOE components.

2.5.4 Identification and Authentication

Each TOE model provides a local password authentication mechanism for the GUI, local CLI, and remote CLI that obscures password upon entry. Users accessing the remote CLI on each model can also authenticate using their SSH public key. The TOE models also enforce password length requirements and will lock users out due to too many failed authentication attempts. The only function available to an unauthenticated user is the ability to acknowledge a warning banner.

The TOE uses X.509 certificates to authenticate servers that it connects to over TLS. This includes each model connecting to the syslog server as well as EX3000 and EX4000 verifying the other TOE component's X.509 certificates when they communicate. The TSF determines the validity of the certificates by confirming the validity of the certificate chain and verifying that the certificate chain ends in a trusted Certificate Authority (CA). The TSF connects with an OCSP Responder through HTTP to confirm certificate validity and revocation. The TSF can generate a Certificate Request that contains the "Common Name" and public key.

2.5.5 Security Management

Each model of the TOE can be administered locally and remotely and uses role based access control (RBAC) to restrict privileges to authorized roles. The Security Administrator roles on the CLI are the "Exabeam user" role and the root account (can authenticate via the local CLI only). For the GUI, users with the "Administrator" role are considered the Security Administrators.

2.5.6 Protection of the TSF

The TOE stores passwords in a variety of locations on each model depending on their use and encryption. They cannot be viewed by any user regardless of the user's role. Additionally, pre-shared keys, symmetric keys, and private keys cannot be accessed in plaintext form by any user. There is software clock backed by an underlying hardware clock on each model that is used for accurate timekeeping and is set by the Security Administrator. Power-on self-tests are executed automatically on each TOE model during the boot process which includes verifying the TOE software's and cryptographic module's integrity. The TOE's DRBG also performs its own health tests.

The version of the software installed on each model is verified via the GUI. The Exabeam user will SCP push (over SSH) the software package from their management workstation to each TOE component and then will run the commands to update the TOE component's software. The software update process includes two different verifications of a SHA-256 public hash.

2.5.7 TOE Access

The TOE models display a configurable warning banner on each user interface prior to the user authenticating to that interface. The TOE components can terminate local CLI, remote CLI, and GUI sessions after a specified time period of inactivity. Administrator users have the capability to terminate their own sessions. Once a session has been terminated the TOE requires the user to re-authenticate to establish a new session.

2.5.8 Trusted Path/Channels

The TOE components connect and send data to IT entities via trusted channels. In the evaluated configuration, each model connects to a syslog server via TLS to send audit data for remote storage. TLS is used for the transfer of collected network event data from EX3000 to EX4000. TLS/HTTPS and SSH are used for remote administration of the TOE via the GUI and remote CLI respectively.

3 Conformance Claims

3.1 CC Version

This ST is compliant with Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 4 September 2012.

3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 2 extended to include all applicable NIAP and International interpretations through July 26, 2019.

3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) are conformant to Part 3 to include all applicable NIAP and International interpretations through July 26, 2019.

3.4 PP Claims

This ST claims exact conformance to the following Protection Profiles:

- collaboration Protection Profile for Network Devices, version 2.0 + Errata 20180314 [NDcPP]

The following is the list of NIAP Technical Decisions that are applicable to the ST/TOE and a summary of their impact:

TD #	Title	Changes			Analysis to this evaluation	
		SFR	AA	Notes	NA	Reason
TD0425	NIT Technical Decision for Cut-and-paste Error for Guidance AA		X			AA: AGD TD has been implemented
TD0423	NIT Technical Decision for Clarification about application of RfI#201726rev2			X		TD has been implemented
TD0412	NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy		X			AA: Test TD has been implemented
TD0411	NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 - Server and client side seem to be confused		X		X	AA: Test Not Applicable: Not claiming FCS_SSHC_EXT.1 in this evaluation
TD0410	NIT technical decision for Redundant assurance activities associated with FAU_GEN.1		X			AA: AGD TD has been implemented
TD0409	NIT decision for Applicability of FIA_AFL.1 to key-based SSH authentication			X		TD has been implemented
TD0408	NIT Technical Decision for local vs. remote administrator accounts	X	X	X		AA: TSS and AGD TD has been implemented

TD0407	NIT Technical Decision for handling Certification of Cloud Deployments			X	X	Not Applicable: The TOE is not a cloud deployment
TD0402	NIT Technical Decision for RSA-based FCS_CKM.2 Selection	X		X		AA: Test TD has been implemented
TD0401	NIT Technical Decision for Reliance on external servers to meet SFRs			X		TD has been implemented
TD0400	NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment			X		TD has been implemented
TD0399	NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)			X	X	Not Applicable: This evaluation does not claim the use of CRLs
TD0398	NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR	X		X		TD has been implemented
TD0397	NIT Technical Decision for Fixing AES-CTR Mode Tests		X			AA: Test TD has been implemented
TD0396	NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2		X			AA: Test TD has been implemented
TD0395	NIT Technical Decision for Different Handling of TLS1.1 and TLS1.2		X			AA: Test TD has been implemented
TD0394	NIT Technical Decision for Audit of Management Activities related to Cryptographic Keys			X		TD has been implemented
TD0343	NIT Technical Decision for Updating FCS_IPSEC_EXT.1.14 Tests	X	X		X	AA: TSS, AGD, and Test Not Applicable: Not claiming IPSEC in this evaluation
TD0342	NIT Technical Decision for TLS and DTLS Server Tests		X			AA: Test TD has been implemented
TD0341	NIT Technical Decision for TLS wildcard checking			X		TD has been implemented
TD0340	NIT Technical Decision for Handling of the basicConstraints extension in CA and leaf certificates	X				TD has been implemented
TD0339	NIT Technical Decision for Making password-based authentication optional in FCS_SSHS_EXT.1.2	X	X	X		AA: TSS and Test TD has been implemented
TD0338	NIT Technical Decision for Access Banner Verification		X			AA: TSS TD has been implemented
TD0337	NIT Technical Decision for Selections in FCS_SSH*_EXT.1.6	X	X	X		AA: Test TD has been implemented

						Note: Supersedes TD0260
TD0336	NIT Technical Decision for Audit requirements for FCS_SSH*_EXT.1.8		X			AA: Test TD has been implemented
TD0335	NIT Technical Decision for FCS_DTLS Mandatory Cipher Suites			X		TD has been implemented
TD0334	NIT Technical Decision for Testing SSH when password-based authentication is not supported		X		X	AA: Test Not Applicable: Not claiming SSH as a client in this evaluation
TD0333	NIT Technical Decision for Applicability of FIA_X509_EXT.3	X	X	X		AA: AGD and Test TD has been implemented
TD0324	NIT Technical Decision for Correction of section numbers in SD Table 1		X			AA: FSP Evaluation Activities TD has been implemented
TD0323	NIT Technical Decision for DTLS server testing - Empty Certificate Authorities list		X		X	AA: Test Not Applicable: Not claiming DTLS in this evaluation
TD0322	NIT Technical Decision for TLS server testing - Empty Certificate Authorities list		X			AA: Test TD has been implemented Note: Supersedes TD0262
TD0321	Protection of NTP communications			X		TD has been implemented
TD0291	NIT technical decision for DH14 and FCS_CKM.1	X	X			AA: Test – this is a note on the test AA
TD0290	NIT technical decision for physical interruption of trusted path/channel.		X			AA: TSS and Test TD has been implemented
TD0289	NIT technical decision for FCS_TLSC_EXT.x.1 Test 5e		X			AA: Test TD has been implemented
TD0281	NIT Technical Decision for Testing both thresholds for SSH rekey		X			AA: Test – this is a note on the test AA TD has been implemented
TD0259	NIT Technical Decision for Support for X509 ssh rsa authentication IAW RFC 6187	X		X		TD has been implemented
TD0257	NIT Technical Decision for Updating FCS_DTLSC_EXT.x.2/FCS_TLSC_EXT.x.2 Tests 1-4		X			AA: Test TD has been implemented
TD0256	NIT Technical Decision for Handling of TLS connections with and without mutual authentication		X			AA: Test TD has been implemented

TD0228	NIT Technical Decision for CA certificates - basicConstraints validation		X			AA: Test TD has been implemented

Table 7: NDcPP Technical Decisions

3.5 Package Claims

The TOE claims exact conformance to the collaborative Protection Profile for Network Devices, version 2.0 + Errata 20180314 [NDcPP] which is conformant with CC Part 3.

The TOE claims the following Selection-Based SFRs that are defined in the appendices of the claimed PP:

- FCS_HTTPS_EXT.1
- FCS_SSHS_EXT.1
- FCS_TLSC_EXT.1
- FCS_TLSC_EXT.2
- FCS_TLSS_EXT.1
- FCS_TLSS_EXT.2
- FIA_X509_EXT.1/Rev
- FIA_X509_EXT.2
- FIA_X509_EXT.3

The TOE claims the following Optional SFRs that are defined in the appendices of the claimed PP:

- FAU_STG.1
- FIA_X509_EXT.1/ITT
- FMT_MTD.1/CryptoKeys
- FPT_ITT.1
- FCO_CPC_EXT.1

This does not violate the notion of exact conformance because the PP specifically indicates these as allowable options and provides both the ST author and evaluation laboratory with instructions on how these claims are to be documented and evaluated.

3.6 Package Name Conformant or Package Name Augmented

This ST and TOE are in exact conformance with the NDcPP.

3.7 Conformance Claim Rationale

The NDcPP states the following: “This is a Collaborative Protection Profile (cPP) whose Target of Evaluation (TOE) is a network device. It provides a minimal set of security requirements expected by all network devices that target the mitigation of a set of defined threats. This baseline set of requirements will be built upon by future cPPs to provide an overall set of security solutions for networks up to carrier and enterprise scale. A network device in the context of this cPP is a device composed of both hardware and software that is connected to the network and has an infrastructure role within the network”.

The TOE is a network device composed of hardware and software that is connected to the network.

4 Security Problem Definition

4.1 Threats

This section identifies the threats against the TOE. These threats have been taken from the NDcPP.

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

	device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the network device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

Table 8: Threats

4.2 Organizational Security Policies

This section identifies the organizational security policies which are expected to be implemented by an organization that deploys the TOE. These policies have been taken from the NDcPP.

Policy	Policy Definition
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.

Table 9: TOE Organizational Security Policy

4.3 Assumptions

The specific conditions listed in this section are assumed to exist in the TOE's Operational Environment. These assumptions have been taken from the NDcPP.

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

A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.
A.REGULAR_UPDATES	The network device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.
A.COMPONENTS_RUNNING	For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Table 10: Assumptions

4.4 Security Objectives

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

Note: This section only discusses environmental objectives because the NDcPP does not contain TOE objectives.

4.4.1 TOE Security Objectives

4.4.2 Security Objectives for the Operational Environment

The TOE's operational environment must satisfy the following objectives:

Objective	Objective Definition

OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.COMPONENTS_RUNNING	For distributed TOEs the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Table 11: TOE Operational Environment Objectives

4.5 Security Problem Definition Rationale

The assumptions, threats, OSPs, and objectives that are defined in this ST represent the assumptions, threats, OSPs, and objectives that are specified in the Protection Profile to which the TOE claims conformance. The associated mappings of assumptions to environmental objectives, SFRs to TOE objectives, and OSPs and objectives to threats are therefore identical to the mappings that are specified in the claimed Protection Profile.

5 Extended Components Definition

5.1 Extended Security Functional Requirements

The extended Security Functional Requirements that are claimed in this ST are taken directly from the PP to which the ST and TOE claim conformance. These extended components are formally defined in the PP in which their usage is required. Therefore the “Extended” used in SFR component name will be dropped.

5.2 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

6 Security Functional Requirements

6.1 Conventions

The CC permits four functional component operations—assignment, refinement, selection, and iteration—to be performed on functional requirements. This ST will highlight the operations in the following manner:

- **Assignment:** allows the specification of an identified parameter. Indicated with *italicized* text.
- **Refinement:** allows the addition of details. Indicated with **bold** text.
- **Selection:** allows the specification of one or more elements from a list. Indicated with underlined text.
- **Iteration:** allows a component to be used more than once with varying operations. Indicated with a sequential number in parentheses following the element number of the iterated SFR and/or separated by a "/" with a notation that references the function for which the iteration is used, e.g. "/ManualUpdate" for an SFR that relates to update functionality.

When multiple operations are combined, such as an assignment that is provided as an option within a selection or refinement, a combination of the text formatting is used.

If SFR text is reproduced verbatim from text that was formatted in a claimed PP (such as if the PP's instantiation of the SFR has a refinement or a completed assignment), the formatting is not preserved. This is so that the reader can identify the operations that are performed by the ST author as opposed to the PP author.

6.2 Security Functional Requirements Summary

The following table lists the SFRs claimed by the TOE:

Class Name	Component Identification	Component Name
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
	FAU_STG_EXT.1	Protected Audit Event Storage
	FAU_STG.1	Protected Audit Storage
Communication	FCO_CPC_EXT.1	Communication Partner Control
Cryptographic Support	FCS_CKM.1	Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM.2	Cryptographic Key Establishment
	FCS_CKM.4	Cryptographic Key Destruction

Class Name	Component Identification	Component Name
	FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
	FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
	FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
	FCS_RBG_EXT.1	Random Bit Generation
	FCS_HTTPS_EXT.1	HTTPS Protocol
	FCS_SSHS_EXT.1	SSH Server Protocol
	FCS_TLSC_EXT.1	TLS Client Protocol
	FCS_TLSC_EXT.2	TLS Client Protocol with authentication
	FCS_TLSS_EXT.1	TLS Server Protocol
	FCS_TLSS_EXT.2	TLS Server Protocol with mutual authentication
Identification and Authentication	FIA_AFL.1	Authentication Failure Management
	FIA_PMG_EXT.1	Password Management
	FIA_UAU_EXT.2	Password-based Authentication Mechanism
	FIA_UAU.7	Protected Authentication Feedback
	FIA_UIA_EXT.1	User Identification and Authentication
	FIA_X509_EXT.1/ITT	X.509 Certificate Validation
	FIA_X509_EXT.1/Rev	X.509 Certificate Revocation
	FIA_X509_EXT.2	X.509 Certificate Validation
	FIA_X509_EXT.3	X.509 Certificate Validation
Security Management	FMT_MOF.1/ManualUpdate	Management of Security Functions Behavior
	FMT_MTD.1/CoreData	Management of TSF Data

Class Name	Component Identification	Component Name
	FMT_MTD.1/CryptoKeys	Management of TSF Data
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.2	Restrictions on Security Roles
Protection of the TSF	FPT_ITT.1	Basic Internal TSF Data Transfer Protection
	FPT_APW_EXT.1	Protection of Administrator Passwords
	FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	FPT_STM_EXT.1	Reliable Time Stamps
	FPT_TST_EXT.1	TSF Testing
	FPT_TUD_EXT.1	Trusted Update
TOE Access	FTA_SSL_EXT.1	TSF-initiated Session Locking
	FTA_SSL.3	TSF-initiated Termination
	FTA_SSL.4	User-initiated Termination
	FTA_TAB.1	Default TOE Access Banners
Trusted Path /Channels	FTP_ITC.1	Inter-TSF Trusted Channel
	FTP_TRP.1/Admin	Trusted Path

Table 12: Security Functional Requirements for the TOE

6.3 Security Functional Requirements

6.3.1 Class FAU: Security Audit

6.3.1.1 *FAU_GEN.1 Audit Data Generation*

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:
   - Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators)
   - Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed)
   - Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged)
   - Resetting passwords (name of related user account shall be logged)
   - [no other actions];
d) Specifically defined auditable events listed in Table **13**.

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of Table **13**.

Requirements	Auditable Events	Additional Audit Record Contents
FCO_CPC_EXT.1	Enabling communications between a pair of components. Disabling communications between a pair of components.	Identities of the endpoints pairs enabled or disabled.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure.
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSC_EXT.2	Failure to establish a TLS Session	Reason for failure.
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.2	Failure to establish a TLS Session	Reason for failure.

Requirements	Auditable Events	Additional Audit Record Contents
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded	Origin of the attempt (e.g. IP address).
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_X509_EXT.1/ITT	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	All management activities of the TSF	None
FMT_MTD.1/CryptoKeys	Management of cryptographic keys	None.
FPT_ITT.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FPT_STM_EXT.1	Discontinuous changes to time – either Administrator actuated or changed via an automated process.	For discontinuous changes to time: the old and new values for the time. Origin of the attempt to change time for success and failure (e.g. IP address)
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FTA_SSL_EXT.1	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failures of the trusted path functions.	Identification of the claimed user identity.

Table 13: Auditable Events

6.3.1.2 *FAU_GEN.2 User Identity Association*

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

6.3.1.3 *FAU_STG_EXT.1 Protected Audit Event Storage*

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

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself.

FAU_STG_EXT.1.3 The TSF shall [overwrite previous audit data records according to the following rule: [*delete the archived file, move previously active file to archived, and create a new active file*]] when the local storage space for audit data is full.

6.3.1.4 *FAU_STG.1 Protected Audit Trail Storage*

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

FAU_STG.1.2 The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

6.3.2 Class FCO: Communication

6.3.2.1 *FCO_CPC_EXT.1 Communication Partner Control*

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

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

- No channel].

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

6.3.3 Class FCS: Cryptographic Support

6.3.3.1 *FCS_CKM.1 Cryptographic Key Generation*

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

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;

- FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1
- FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3].[1]

6.3.3.2 *FCS_CKM.2 Cryptographic Key Establishment*

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

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1;
- Finite field-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";
- Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3].[2]

6.3.3.3 *FCS_CKM.4 Cryptographic Key Destruction*

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

- For plaintext keys in volatile storage, the destruction shall be executed by a [destruction of reference to the key directly followed by a request for garbage collection];
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [
    - instructs a part of the TSF to destroy the abstraction that represents the key]

that meets the following: No Standard.

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

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

[1] TD0291
[2] TD0402

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

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

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

that meet the following: [

- For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,

].

6.3.3.6 **FCS_COP.*1/Hash*** ***Cryptographic Operation (Hash Algorithm)***

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

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

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

6.3.3.8 *FCS_RBG_EXT.1 Random Bit Generation*

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

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*4*] software-based noise sources] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and CSPs that it will generate.

6.3.3.9 *FCS_HTTPS_EXT.1 HTTPS Protocol*

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

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

6.3.3.10 *FCS_SSHS_EXT.1 SSH Server Protocol*

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs [4251, 4252, 4253, 4254, 6668].[3]

FCS_SSHS_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].[4]

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*35,000*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [aes256-cbc].[5]

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa] as its public key algorithm(s) and rejects all other public key algorithms.[6]

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha2-256, hmac-sha2-512] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).[7]

FCS_SSHS_EXT.1.7 The TSF shall ensure that [diffie-hellman-group14-sha1] and [no other methods] are the only allowed key exchange methods used for the SSH protocol.

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

6.3.3.11 *FCS_TLSC_EXT.1 TLS Client Protocol*

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

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246].

[3] TD0398
[4] TD0339
[5] TD0337
[6] TD0259
[7] TD0337

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

FCS_TLSC_EXT.1.3 The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [not establish the connection].

FCS_TLSC_EXT.1.4 The TSF shall [not present the Supported Elliptic Curves Extension] in the Client Hello.

6.3.3.12 *FCS_TLSC_EXT.2 TLS Client Protocol with authentication*

FCS_TLSC_EXT.2.1 The TSF shall implement [TLS 1.2 (RFC 5246)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246].

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

FCS_TLSC_EXT.2.3 The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [not establish the connection].

FCS_TLSC_EXT.2.4 The TSF shall [not present the Supported Elliptic Curves Extension] in the Client Hello.

FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

6.3.3.13 *FCS_TLSS_EXT.1 TLS Server Protocol*

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

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246].

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

FCS_TLSS_EXT.1.3 The TSF shall [perform RSA key establishment with key size [2048 bits]; generate Diffie-Hellman parameters of size [2048 bits]].

6.3.3.14 *FCS_TLSS_EXT.2 TLS Server Protocol with mutual authentication*

FCS_TLSS_EXT.2.1 The TSF shall implement [TLS 1.2 (RFC 5246)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246].

FCS_TLSS_EXT.2.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [TLS 1.1].

FCS_TLSS_EXT.2.3 The TSF shall [perform RSA key establishment with key size [2048 bits]; generate Diffie-Hellman parameters of size [2048 bits]].

FCS_TLSS_EXT.2.4 The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.5 The TSF shall not establish a trusted channel if the client certificate is invalid. If the client certificate is deemed invalid, then the TSF shall [not establish the connection].

FCS_TLSS_EXT.2.6 The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the client.

6.3.4 Class FIA: Identification and Authentication

6.3.4.1 *FIA_AFL.1 Authentication Failure Management*

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1 to 20*] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.[8]

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until [*unlocking the offending Administrator's account*] is taken by an Administrator; prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].[9]

[8] TD0408
[9] TD0408

6.3.4.2 *FIA_X509_EXT.1/ITT X.509 Certificate Validation*

FIA_X509_EXT.1.1/ITT The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of two certificates.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.[10]
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 6960]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

6.3.4.3 *FIA_X509_EXT.1/Rev X.509 Certificate Revocation*

FIA_X509_EXT.1.1/Rev The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of three certificates.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.[11]
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 6960]

[10] TD0340
[11] TD0340

- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

6.3.4.4 *FIA_X509_EXT.2* *X.509 Certificate Validation*

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

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

6.3.4.5 *FIA_X509_EXT.3* *X.509 Certificate Validation*

FIA_X509_EXT.3.1 The TSF shall generate a Certification Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [Common Name].[12]

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

6.3.4.6 *FIA_PMG_EXT.1* *Password Management*

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”]

b) Minimum password length shall be configurable to [*1 character for the GUI, 5 characters for the CLI*] and [*15 characters for both the GUI and CLI*].

[12] TD0333

6.3.4.7 *FIA_UAU_EXT.2 Password-based Authentication Mechanism*

FIA_UAU_EXT.2.1 The TSF shall provide a local [password-based, SSH public key-based] authentication mechanism to perform local administrative user authentication.[13]

6.3.4.8 *FIA_UAU.7 Protected Authentication Feedback*

FIA_UAU.7.1 The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress at the local console.

6.3.4.9 *FIA_UIA_EXT.1 User Identification and Authentication*

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

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

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

6.3.5 Class FMT: Security Management

6.3.5.1 *FMT_MOF.1/ManualUpdate Management of Security Functions Behavior*

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

6.3.5.2 *FMT_MTD.1/CoreData Management of TSF Data*

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

6.3.5.3 *FMT_MTD.1/CryptoKeys Management of TSF Data*

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to manage the cryptographic keys to the Security Administrators.

6.3.5.4 *FMT_SMF.1 Specification of Management Functions*

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

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;

[13] TD0408

- Ability to update the TOE, and to verify the updates using [hash comparison] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [
  - Ability to configure the cryptographic functionality;
  - Ability to configure the interaction between TOE components;
  - Ability to re-enable an Administrator account;
  - Ability to set the time which is used for time-stamps]

6.3.5.5 *FMT_SMR.2 Restrictions on Security Roles*

FMT_SMR.2.1 The TSF shall maintain the roles:

- Security Administrator.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions**:**

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

are satisfied.

6.3.6 Class FPT: Protection of the TSF

6.3.6.1 *FPT_ITT.1 Basic Internal TSF Data Transfer Protection*

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

6.3.6.2 *FPT_APW_EXT.1 Protection of Administrator Passwords*

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

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

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

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

6.3.6.4 *FPT_STM_EXT.1 Reliable Time Stamps*

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

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

6.3.6.5 *FPT_TST_EXT.1 TSF Testing*

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [during initial start-up (on power on)] to demonstrate the correct operation of the TSF: [*software image integrity test, cryptographic module integrity test, DRBG Known Answer Test, Continuous Random Number Generator test, SP 800-90B health tests*]

6.3.6.6 *FPT_TUD_EXT.1 Trusted Update*

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

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

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

6.3.7 Class FTA: TOE Access

6.3.7.1 *FTA_SSL_EXT.1 TSF-initiated Session Locking*

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions,

- [terminate the session]

after a Security Administrator-specified time period of inactivity.

6.3.7.2 *FTA_SSL.3 TSF-initiated Termination*

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

6.3.7.3 *FTA_SSL.4 User-initiated Termination*

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

6.3.7.4 *FTA_TAB.1 Default TOE Access Banners*

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

6.3.8 **Class FTP: Trusted Path/Channels**

6.3.8.1 ***FTP_ITC.1 Inter-TSF Trusted Channel***

FTP_ITC.1.1	The TSF shall be capable of using [TLS] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [no other capabilities] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.
FTP_ITC.1.2	The TSF shall permit the TSF, or the authorized IT entities to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for [*audit records*].

6.3.8.2 ***FTP_TRP.1/Admin Trusted Path***

FTP_TRP.1.1/Admin	The TSF shall be capable of using [SSH, TLS, HTTPS] provide a communication path between itself and authorized remote Administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the channel data**.**
FTP_TRP.1.2/Admin	The TSF shall permit remote Administrators to initiate communication via the trusted path.
FTP_TRP.1.3/Admin	The TSF shall require the use of the trusted path for initial Administrator authentication and all remote administration actions.

6.4 Statement of Security Functional Requirements Consistency

The Security Functional Requirements included in the ST represent all required SFRs specified in the claimed PP as well as a subset of the Selection-Based and Optional SFRs. All hierarchical relationships, dependencies, and unfulfilled dependency rationales in the ST are considered to be identical to those that are defined in the claimed PP.

7 Security Assurance Requirements

This section identifies the Security Assurance Requirements (SARs) that are claimed for the TOE. The SARs which are claimed are in exact conformance with the NDcPP.

7.1 Class ADV: Development

7.1.1 Basic Functional Specification (ADV_FSP.1)

7.1.1.1 *Developer action elements:*

ADV_FSP.1.1D

The developer shall provide a functional specification.

ADV_FSP.1.2D

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

7.1.1.2 *Content and presentation elements:*

ADV_FSP.1.1C

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2C

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3C

The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C

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

7.1.1.3 *Evaluator action elements:*

ADV_ FSP.1.1E

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

ADV_ FSP.1.2E

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

7.2 Class AGD: Guidance Documentation

7.2.1 Operational User Guidance (AGD_OPE.1)

7.2.1.1 *Developer action elements:*

AGD_OPE.1.1D

The developer shall provide operational user guidance.

7.2.1.2 *Content and presentation elements:*

AGD_OPE.1.1C

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

AGD_OPE.1.2C

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

AGD_OPE.1.3C

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

AGD_OPE.1.4C

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

AGD_OPE.1.5C

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

AGD_OPE.1.6C

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

AGD_OPE.1.7C

The operational user guidance shall be clear and reasonable.

7.2.1.3 *Evaluator action elements:*

AGD_OPE.1.1E

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

7.2.2 Preparative Procedures (AGD_PRE.1)

7.2.2.1 *Developer action elements:*

AGD_PRE.1.1D

The developer shall provide the TOE including its preparative procedures.

7.2.2.2 *Content and presentation elements:*

AGD_ PRE.1.1C

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

AGD_ PRE.1.2C

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

7.2.2.3 *Evaluator action elements:*

AGD_ PRE.1.1E

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

AGD_ PRE.1.2E

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

7.3 Class ALC: Life Cycle Supports

7.3.1 Labeling of the TOE (ALC_CMC.1)

7.3.1.1 *Developer action elements:*

ALC_CMC.1.1D

The developer shall provide the TOE and a reference for the TOE.

7.3.1.2 *Content and presentation elements:*

ALC_CMC.1.1C

The TOE shall be labeled with its unique reference.

7.3.1.3 *Evaluator action elements:*

ALC_CMC.1.1E

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

7.3.2 TOE CM Coverage (ALC_CMS.1)

7.3.2.1 *Developer action elements:*

ALC_CMS.1.1D

The developer shall provide a configuration list for the TOE.

7.3.2.2 *Content and presentation elements:*

ALC_CMS.1.1C

The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2C

The configuration list shall uniquely identify the configuration items.

7.3.2.3 *Evaluator action elements:*

ALC_CMS.1.1E

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

7.4 Class ATE: Tests

7.4.1 Independent Testing - Conformance (ATE_IND.1)

7.4.1.1 *Developer action elements:*

ATE_IND.1.1D

The developer shall provide the TOE for testing.

7.4.1.2 *Content and presentation elements:*

ATE_IND.1.1C

The TOE shall be suitable for testing.

7.4.1.3 *Evaluator action elements:*

ATE_IND.1.1E

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

ATE_IND.1.2E

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

7.5 Class AVA: Vulnerability Assessment

7.5.1 Vulnerability Survey (AVA_VAN.1)

7.5.1.1 *Developer action elements:*

AVA_VAN.1.1D

The developer shall provide the TOE for testing.

7.5.1.2 *Content and presentation elements:*

AVA_VAN.1.1C

The TOE shall be suitable for testing.

7.5.1.3 *Evaluator action elements:*

AVA_VAN.1.1E

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

AVA_VAN.1.2E

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

AVA_VAN.1.3E

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

8 TOE Summary Specification

The following sections identify the security functions of the TOE and describe how the TSF meets each claimed SFR. They include Security Audit, Communication, Cryptographic Support, Identification and Authentication, Security Management, Protection of the TSF, TOE Access and Trusted Path / Channels. The following table defines which distributed TOE component(s) perform the capabilities described by the SFR.

Requirement	EX3000	EX4000
FAU_GEN.1	X	X
FAU_GEN.2	X	X
FAU_STG.1	X	X
FAU_STG_EXT.1	X	X
FCO_CPC_EXT.1	X	X
FCS_CKM.1	X	X
FCS_CKM.2	X	X
FCS_CKM.4	X	X
FCS_COP.1/DataEncryption	X	X
FCS_COP.1/SigGen	X	X
FCS_COP.1/Hash	X	X
FCS_COP.1/KeyedHash	X	X
FCS_RBG_EXT.1	X	X
FCS_HTTPS_EXT.1	X	X
FCS_SSHS_EXT.1	X	X
FCS_TLSC_EXT.1	X	X
FCS_TLSC_EXT.2	X	
FCS_TLSS_EXT.1	X	X
FCS_TLSS_EXT.2		X
FIA_AFL.1	X	X
FIA_PMG_EXT.1	X	X
FIA_UIA_EXT.1	X	X
FIA_UAU_EXT.2	X	X
FIA_UAU.7	X	X
FIA_X509_EXT.1/Rev	X	X
FIA_X509_EXT.1/ITT	X	X
FIA_X509_EXT.2	X	X
FIA_X509_EXT.3	X	X
FMT_SMF.1	X	X
FMT_SMR.2	X	X
FMT_MOF.1/ManualUpdate	X	X
FMT_MTD.1/CoreData	X	X
FMT_MTD.1/CryptoKeys	X	X
FPT_ITT.1	X	X
FPT_SKP_EXT.1	X	X
FPT_APW_EXT.1	X	X
FPT_TST_EXT.1	X	X
FPT_TUD_EXT.1	X	X
FPT_STM_EXT.1	X	X
FTA_SSL_EXT.1	X	X
FTA_SSL.3	X	X
FTA_SSL.4	X	X

FTA_TAB.1	X	X
FTP_ITC.1	X	X
FTP_TRP.1/Admin	X	X

Table 14: SFR and TOE Component Mapping

8.1 Security Audit

8.1.1 FAU_GEN.1 and FAU GEN.2:

The TOE has the mechanisms to automatically generate audit records based on the behavior that occurs within the TSF. The audit records include all successful and unsuccessful management actions by all authorized users of the TOE. The startup and shutdown of the TOE's audit functionality is synonymous with the startup and shutdown of the TOE. In the evaluated configuration, the audit functions of the TOE are provided by rsyslog and the audit functions can be enabled or disabled by the root user on the local CLI. When the TOE's audit functions are enabled or disabled, the TOE will generate an audit record of this occurring. Each audit record contains identifying information including the date and time the event occurred, the type of event, the subject identity of the event, and the outcome of the event. Each TOE component collects audit records in rsysreceived.log on that TOE component. The audit logs are then securely sent to a syslog server in the operational environment over TLS.

Table 15 identifies the auditable events that are inclusive to the PP and which TOE component will record the event.

Requirement	EX 3000	EX 4000	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	X	X	Start-up and shut-down of the audit functions	None
FAU_GEN.1	X	X	Administrative login and logout	Name of user account shall be logged if individual user accounts are required for Administrators
FAU_GEN.1	X	X	Changes to TSF data related to configuration changes	In addition to the information that a change occurred it shall be logged what has been changed
FAU_GEN.1	X	X	Generating/import of, changing, or deleting of cryptographic keys	In addition to the action itself a unique key name or key reference shall be logged
FAU_GEN.1	X	X	Resetting passwords	Name of related user account shall be logged
FCO_CPC_EXT.1	X	X	Enabling communications between a pair of components. Disabling communications between a pair of components.	Identities of the endpoints pairs enabled or disabled.
FCS_HTTPS_EXT.1	X	X	Failure to establish a HTTPS Session.	Reason for failure
FCS_SSHS_EXT.1	X	X	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.1	X	X	Failure to establish a TLS Session	Reason for failure

Requirement	EX 3000	EX 4000	Auditable Events	Additional Audit Record Contents
FCS_TLSC_EXT.2	X		Failure to establish a TLS Session	Reason for failure.
FCS_TLSS_EXT.1	X	X	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.2		X	Failure to establish a TLS Session	Reason for failure.
FIA_AFL.1	X	X	Unsuccessful login attempts limit is met or exceeded	Origin of the attempt (e.g. IP address).
FIA_UIA_EXT.1	X	X	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	X	X	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_X509_EXT.1/ITT	X	X	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.1/Rev	X	X	Unsuccessful attempt to validate a certificate	Reason for failure
FMT_MOF.1/Manual Update	X	X	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	X	X	All management activities of the TSF	None.
FMT_MTD.1/CryptoKeys	X	X	Management of cryptographic keys	None.
FPT_ITT.1	X	X	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FPT_STM_EXT.1	X	X	Discontinuous changes to time – either Administrator actuated or changed via an automated process.	For discontinuous changes to time: the old and new values for the time. Origin of the attempt to change time for success and failure (e.g. IP address)
FPT_TUD_EXT.1	X	X	Initiation of update; result of the update attempt (success or failure)	None.
FTA_SSL_EXT.1	X	X	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	X	X	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	X	X	The termination of an interactive session.	None.
FTP_ITC.1	X	X	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	X	X	Initiation of the trusted path. Termination of the trusted path. Failures of the trusted path functions.	Identification of the claimed user identity

Table 15: Audit for TOE Components

Audit records are created when the administrator performs each of the management functions listed above via the GUI and the CLI (local and remote), when an external operational environment component is connecting with the TOE, and when the TOE components are connecting with each other. Each audit record provides a timestamp, subject identity, defines the type of event, and identifies if the event was successful or failed. The TOE also records additional information as specified by the right column of Table 15. For example, when generating/import of, changing, or deleting of cryptographic keys the audit record will contain the filename to identify the key.

8.1.2 FAU_STG.1 and FAU_STG_EXT.1:

Each TOE component stores its audit records in its own rsysreceived.log. Simultaneously, in the evaluated configuration each TOE component sends its audit records securely to a syslog server over TLS which occurs in real-time. This requires the Exabeam user to configure via the CLI (local or remote) the syslog server to which the audit records will be sent and only one syslog server can be configured on each TOE component. Each TOE component handles its own audit processes and does not receive audit records from the other TOE component. Note the network event data that EX3000 sends to EX4000 does not contain EX3000's audit records.

The maximum allocated space for rsysreceived.log is 21GB. The rsysreceived.log function has 2 log files and each log file's size is 10.5GB. When both audit log files are full, rsysreceived.log will roll the audit log files by deleting the archived log file, turning the active log file into the archived file, and creating a new active log file for rsysreceived.log; to which new audit records are written.

Only the Exabeam user (local and remote CLI) and root user (local CLI only) can delete the audit logs and only the root user (local CLI only) can modify the audit logs. This is enforced by the TOE's permissions assigned to its users and what management activities can be performed over its interfaces. The audit functionality starts automatically with the TOE's boot up process. In the evaluated configuration, the audit functions of the TOE are provided by rsyslog and the audit functions can be enabled or disabled by the root user on the local CLI.

8.2 Communication

8.2.1 FCO_CPC_EXT.1:

The TOE requires a Security Administrator to enable communication between TOE components. The registration process is accomplished through administration only and the initial TLS connection between components is the one used as part of the normal internal TSF communications. The purpose of the connection between EX3000 and EX4000 is for EX3000 to send network events it collects to EX4000 for the SMP's primary purpose.

The EX4000 component must be configured first by the Security Administrator to ensure that it is ready to receive the collected network events as soon as the EX3000 has been configured to send them. The Security Administrator will authenticate to the local CLI of the EX4000 and will execute the Ansible configuration process. Ansible will then take the Security Administrator through the automated process that will perform the initial configuration of this TOE component, including its Advanced Analytics and Incident Responder software. Once the Ansible process is complete, the Security Administrator will need to replace the self-signed certificate generated by Ansible with a CA signed certificate for EX4000. The

Security Administrator accomplishes this over the remote CLI using the process described in Section 8.4.5.

The Security Administrator will then authenticate to the local CLI of the EX3000 and will execute the Ansible configuration process. Ansible will then take the Security Administrator through the automated process that will perform the initial configuration of this TOE component, including its Data Lake software. Once the Ansible process is complete, the Security Administrator will need to replace the self-signed certificate generated by Ansible with a CA signed certificate for EX3000. The Security Administrator accomplishes this over the remote CLI using the process described in Section 8.4.5. The Security Administrator will then configure the EX3000 to send its collected network event data to the EX4000 by specifying the IP address of the EX4000 and that the transportation method will be over TLS.

Once configuration is complete and there is collected network event data to be sent, the EX3000 will initiate a TLS connection to EX4000 as specified in Sections 8.3.10 and 8.3.11. This includes EX3000 and the EX4000 verifying the other TOE component's certificate during the TLS handshake as specified in Section 8.4.5. The Security Administrator can disable the communication between the TOE components by authenticating to either the local CLI or remote CLI of the EX3000 and unconfiguring the EX4000's IP address as the location to send the collected network event data.

8.3 Cryptographic Support

Each TOE component contains its own cryptographic module software called OpenSSL 6.0. The cryptographic module is the same software on both components and performs the functionality described within this section the same; except when discussing inter-TOE communications where EX3000 is a TLS client and EX4000 is a TLS server.

Table 6 in section 2.5.2 contains the CAVP algorithm certificates for the cryptographic module implemented in the TOE.

8.3.1 **FCS_CKM.1:**

The TOE implements a FIPS PUB 186-4 conformant key generation mechanism for RSA key generation schemes for establishing TLS server, TLS client, and SSH server connections. Specifically, the TOE complies with the FIPS 186-4 (Digital Signature Standard (DSS) Appendix B.3). This is used to generate the RSA key pairs with a modulus of at least 2048 bits which has an equivalent key strength of 112 bits.

In addition, the TOE implements a FIPS PUB 186-4 conformant key generation mechanism for Diffie-Hellman key establishment schemes with a key size of 2048 bits for TLS server and TLS client connections. Specifically, the TOE complies with the Digital Signature Standard (DSS) Appendix B.1. The TOE also generates RFC 3526, Section 3 conformant key generation mechanism for diffie-hellman-group14-sha1 with a key size of 2048 bits which is used by the TOE when operating as an SSH server.

8.3.2 **FCS_CKM.2:**

The TOE implements a NIST SP 800-56A conformant key establishment mechanism for Diffie-Hellman key establishment schemes. Specifically, the TOE complies with the NIST SP 800-56A Key Agreement Scheme (KAS) without a Key Derivation Function (KDF) which is defined in section 5.6 of the Special Publication. This requirement is met by the Component Validation List (CVL) certificate identified in Table 6. This scheme is used in TLS server and TLS client connections. In addition, the TSF uses diffie-

hellman-group14-sha1 key establishment mechanism with a key size of 2048 bits and in accordance with RFC 3526, Section 3; which is used by the TOE when operating as an SSH server. In addition, the TOE implements RSA key establishment, conformant to RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. This requirement is met by the vendor affirmation. The TOE is able to generate RSA key pairs with a modulus of at least 2048 bits which has an equivalent key strength of 112 bits. This scheme is used in TLS server and TLS client connections. See Table 6 Cryptographic Algorithm Table for certification numbers.

8.3.3 **FCS_CKM.4:**

The following table describes what keys were used, where they are stored, and also how they are destroyed. There are no known instances where key destruction does not happen as defined.

Key Material	**Storage Location**	**Origin**	**Destruction**
SSH Session Keys	RAM	SSH Server/ Client Applications	The reference to the session key is destroyed after it is no longer needed (i.e. connection terminated or re-key) and when the TOE is shutdown or power is lost. The session key's reference is destroyed by the TOE invoking the sshkey_free(struct sshkey *k) function which is followed by a request for garbage collection.
SSH Server Host Private Key	RAM and File system	Generated on platform at set-up of each device	RAM – The reference to the private key is destroyed after it is no longer needed and when the TOE is shutdown or power is lost. The private key's reference is destroyed by the TOE invoking the sshkey_free(struct sshkey *k) function which is followed by a request for garbage collection. File System – The reference to the private key is deleted when the Security Administrator runs the 'rm' command via the CLI. The Security Administrator would perform this action before generating a new key. The TOE will invoke an interface provided by a part of the TSF that instructs a part of the TSF to destroy the abstraction that represents the key (i.e. delete the resource).

TLS Server Host Certificate Private Key	RAM and File system	Generated on platform at set-up or imported after installation	RAM – The reference to the private key is destroyed after it is no longer needed and when the TOE is shutdown or power is lost. The private key's reference is destroyed by the TOE's garbage collection process which will mark the memory as unreferenced, delete the contents of the memory space, and then perform garbage collection. File System – The reference to the private key is deleted when the Security Administrator runs the 'rm' command via the CLI. The Security Administrator would perform this action before generating a new certificate. The TOE will invoke an interface provided by a part of the TSF that instructs a part of the TSF to destroy the abstraction that represents the certificate (i.e. delete the resource).
Diffie-Hellman Shared Secret	RAM	SSH Server/ Client Applications	The reference to the shared secret is destroyed after it is no longer needed (i.e. connection terminated or re-key) and when the TOE is shutdown or power is lost. When being used for SSH, the shared secret's reference is destroyed by the TOE invoking the sshkey_free(struct sshkey *k) function which is followed by a request for garbage collection.
Diffie-Hellman Private Key	RAM	SSH Server/ Client Applications	The reference to the private key is destroyed after it is no longer needed (i.e. connection terminated or re-key) and when the TOE is shutdown or power is lost. When being used for SSH, the private key's reference is destroyed by the TOE invoking the sshkey_free(struct sshkey *k) function which is followed by a request for garbage collection.

Table 16: Cryptographic Materials, Storage, and Destruction Methods

8.3.4 **FCS_COP.1/DataEncryption:**

The TOE provides symmetric encryption and decryption capabilities using AES in CBC mode with 128-bit and 256-bit keys as described in ISO 10116 and GCM mode with 256-bit keys as described in ISO 19772. The TOE provides encryption and decryption in support of SSH and TLS communications. The TOE's AES implementation is validated under CAVP. See Table 6 Cryptographic Algorithm Table for certification numbers. AES in CTR mode with a 256-bit key as described in ISO 10116 is used in the CTR_DRBG(AES).

8.3.5 **FCS_COP.1/SigGen:**

The TOE will provide cryptographic signature services using RSA. RSA is the public-key algorithm used in support of SSH and TLS communications. RSA uses key sizes of 2048 and is validated under CAVP. See Table 6 Cryptographic Algorithm Table for certification numbers.

8.3.6 **FCS_COP.1/Hash:**

The TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384 and SHA-512 as specified in ISO 10118-3:2004. The TOE uses cryptographic hashing services in support of SSH key establishment (SHA-1), HMAC for SSH (SHA-256 and SHA-512), and in support of TLS (SHA-256 and SHA-384). See Table 6 Cryptographic Algorithm Table for certification numbers.

8.3.7 **FCS_COP.1/KeyedHash:**

The TOE provides keyed-hash message authentication services using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with key sizes and message digest sizes of 256 bits, 384 bits, and 512 bits, as specified in ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”. HMAC-SHA-256 uses the hash function SHA-256, has a block size of 512, and uses an output MAC length of 256. HMAC-SHA-384 uses the hash function SHA-384, has a block size of 1024, and uses an output MAC length of 384. HMAC-SHA-512 uses the hash function SHA-512, has a block size of 1024, and uses an output MAC length of 512. HMAC-SHA-256 and HMAC-SHA-384 are used to support TLS communications. HMAC-SHA-256 and HMAC-SHA-512 are used to support SSH communications. See Table 6 Cryptographic Algorithm Table for certification numbers.

8.3.8 **FCS_RBG_EXT.1:**

The TOE implements a counter mode deterministic random bit generator (CTR_DRBG(AES)). The DRBG used by the TOE is in accordance with ISO/IEC 18031:2011. There is no ability to specify the use of an alternative DRBG. The different TOE models uniformly provide 4 software-based noise entropy sources as described in the proprietary entropy specification. The amount of entropy that is collected is based on the function that the DRBG is being used for. In all cases, this amount is greater than or equal to the security strength of the data that is being output which is at least 256 bits. For example, a 256-bit AES key generation operation will collect at least 256 bits of entropy before the DRBG is invoked. The largest key generation operation supported is 2048-bits for both RSA and Diffie-Hellman.

The OpenSSL 6.0 cryptographic module collects entropy from /dev/ random, which is a blocking entropy source. The /dev/random entropy pool is protected by being in kernel memory and is not accessible from user space. The entropy source is described in greater detail in the proprietary Entropy Assessment Report (EAR).

The TOE’s DRBG implementation is validated under CAVP. See Table 6 Cryptographic Algorithm Table for certification numbers.

8.3.9 **FCS_SSHS_EXT.1:**

The TOE acts as an SSH server for remote CLI management sessions on each TOE component. The SSH functionality complies with RFCs 4251, 4252, 4253, 4254, and 6668. The TOE implementation of SSH supports public key-based and password-based authentication using an RSA key of 2048 bits length as

described in RFC 4252, using ssh-rsa as its public key authentication algorithm. The TOE implementation of SSHv2 supports AES-256-CBC for its transport algorithm. Data integrity is assured using HMAC-SHA2-256 and HMAC-SHA2-512 and all other MAC algorithms are rejected. The allowed key exchange methods are diffie-hellman-group14-sha1 and no other key exchange methods. The SSH connection will drop any connection when a packet greater than 35,000 bytes is detected, in accordance with RFC 4253. The SSH connection will rekey before 60 minutes has elapsed or one gigabyte of data has been transmitted using that key, whichever occurs first.

8.3.10 FCS_TLSC_EXT.1 and FCS_TLSC_EXT.2:

The TOE when acting as a TLS client will only support the TLSv1.2 protocol, and will support the following ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246

Each TOE component is a TLS client to connect and secure the following trusted channels and inter-TOE communication:

EX3000 is a TLS client for:

- sending collected network event data to EX4000 (FCS_TLSC_EXT.2)
- transferring audit data to a syslog server (FCS_TLSC_EXT.1)

EX4000 is a TLS client for:

- transferring audit data to a syslog server (FCS_TLSC_EXT.1)

For all of these connections, the TOE performs TLS client authentication and validation of the TLS server-side X.509v3 certificate. For only the EX3000 to EX4000 connection, the TOE performs mutual authentication using X.509v3 certificates where the EX3000 will send its TLS client-side certificate to EX4000 for authentication and validation. Configuring these channels requires the Security Administrator to define the reference identifier of the operational environment servers to which the TOE component will connect and the reference identifier for the EX4000 on the EX3000. Wildcards cannot be defined as part of the reference identifier on the TOE, but the TOE will accept certificates with wildcards specified where they are allowed to be supported. As part of the TLS session establishment, the TOE component will validate the 2048-bit X.509v3 certificate received from the TLS server (syslog server or EX4000) and will only establish the connection if the certificate is valid. The TOE component will also verify the identity of the TLS server in accordance with RFC 6125 by checking that the presented identifier from the certificate, which includes the Common Name (CN) and Subject Alternative Name (SAN), matches the reference identifier defined on the TOE component by the root user via the local CLI. The reference identifier can only be a DNS name; IP addresses are not supported. The TOE does not support certificate pinning or Elliptic Curves Extension.

8.3.11 FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, and FCS_HTTPS_EXT.1:

The TOE when acting as a TLS server will only support the TLSv1.2 protocol, and will support the following ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246

The TSF denies all connections from clients requesting connections dependent on the following SSL 2.0, SSL 3.0, TLS 1.0, and TLS 1.1 protocols. Each TOE component is a TLS server to connect and secure the following trusted paths and inter-TOE communication:

EX3000 is a TLS server for:

- management via the GUI (FCS_TLSS_EXT.1 and FCS_HTTPS_EXT.1)

EX4000 is a TLS server for:

- management via the GUI (FCS_TLSS_EXT.1 and FCS_HTTPS_EXT.1)
- receiving collected network event data for EX3000 (FCS_TLSS_EXT.2)

The HTTPS (HTTP over TLS) provided by each TOE component's GUI is conformant to RFC 2818 per its descriptions of handling HTTPS communications from the server side of the trusted channel connection and it does not enforce TLS mutual authentication.

The EX3000 and the EX4000 perform TLS mutual authentication using X.509v3 certificates. When EX3000 attempts to connect to the EX4000, the EX4000 will present its TLS server-side certificate to the EX3000. EX3000 will then validate the certificate and confirm the EX4000's identity. If validation and/or authentication of the EX4000's server-side certificate is not successful, the EX3000 will end the connection. If validation and authentication of the EX4000's server-side certificate is successful, the EX3000 will then send its TLS client-side certificate to EX4000 for validation and authentication. The EX4000 checks the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate against the expected identifier for the EX3000 (TLS client) configured on the EX4000 by the root user via the local CLI. The expected identifier can only be a DNS name; IP addresses are not supported. This inter-TOE channel will only be established if both certificates are valid and the TOE components are both identified by their counterpart. Each TOE component generates key establishment parameters using either RSA with key sizes of 2048 bits or Diffie-Hellman with key sizes of 2048 bits.

8.4 Identification and Authentication

8.4.1 **FIA_AFL.1:**

Both the EX3000 and EX4000 have GUI and remote CLI interfaces. In the evaluated configuration, the TOE will lock a remote administrative account when an administrator configured number of successive invalid login attempts have been made.

The failed login attempt threshold value for the remote CLI and GUI are configured independently in separate configuration files by the Exabeam user via the local or remote CLI. The threshold for failed login attempts for the remote CLI can be administratively configured to a value between 1 and 20. For the remote CLI's configuration changes to take effect, the Exabeam user must redeploy that TOE component using the Exabeam deployment script. The threshold for failed login attempts for the remote GUI can be

administratively configured to a value between 1 and 20. For the remote GUI's configuration changes to take effect, the Exabeam user must restart the TOE component's web services.

The TOE maintains a counter per username for the number of failed authentication attempts and tracks the time when each failed authentication attempt occurs. If a valid password is provided before the failed attempt threshold value is met, then authentication is granted and the counter resets to zero. If the limit of failed authentication attempts is reached, the account associated with the username will be locked. Once an account is locked, repeated attempts to authenticate with that account will result in displaying the following error message:

- GUI:

```
Number of wrong login attempts exceeded. Your account is
locked. Please contact your Exabeam administrator.
```

- Remote CLI:

```
Permission denied (publickey,keyboard-interactive).
lins-macbook-pro:host1-08-22 lin$ ssh exabeam@192.168.168.74
************************************************************
*****
* This system is for the use of authorized users only.
    *
************************************************************
*****
Account temporarily locked due to 3 failed logins
(2 minutes left to unlock)
```

The user associated with an offending account will be locked out and no authentication attempts will be approved until a Security Administrator manually unlocks the account (remote CLI and GUI accounts) or alternatively, for only a remote CLI account, it can also be unlocked once the lockout time period is reached.

A remote CLI account will be locked for the configured locked time period which is between 120 seconds (2 minutes) and 10,800 seconds (3 hours). The default lockout time period value for the remote CLI is 120 seconds. The remote CLI's lockout time period value is administratively defined by modifying a configuration file by the Exabeam user. For the remote CLI's configuration changes to take effect, the Exabeam user must redeploy that TOE component using the Exabeam deployment script. The GUI does not support a lockout time period.

The Exabeam user can unlock a GUI account via the local or remote CLI. The root user can unlock the Exabeam user account from the local CLI. A GUI user with Administrator privileges can unlock another GUI account by resetting the offending account's password. The root account via the local CLI is not subject to lockout due to authentication failures and thus, authentication failures by remote Security Administrators cannot lead to a situation that prevents all administration of the TOE.

8.4.2 **FIA_PMG_EXT.1:**

The GUI and the CLI on each TOE component has password-based authentication and the passwords can be composed of any combination of upper and lower case letters, numbers and special characters. The accepted special characters for both the GUI and the CLI are: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")." For the GUI, the minimum password length is set by the Exabeam user via the CLI to a value between 1 and 15 characters. Passwords for the CLI have a minimum password length between 5

characters and 15 characters in the evaluated configuration. The minimum password length for the CLI can be configured by the Exabeam user via the CLI.

8.4.3 FIA_UIA_EXT.1 and FIA_UAU_EXT.2:

With respect to TOE security functions as defined by the SFRs, each TOE component performs the same user authentication functions and management functionality except for the differences with inter-TOE communication where the Security Administrators manage the EX3000 as the client and EX4000 as the server within the distributed TOE. Users can authenticate to a TOE component via its CLI or GUI.

The CLI can be accessed remotely through an SSH client or locally with a monitor and keyboard. Prior to authentication, the only pre-authentication service that the TOE allows on the CLI is the display of the standard Linux pre-authentication banner; which can be configured by the Exabeam user through the CLI. The local CLI requires the user to authenticate to the TOE's local authentication mechanism with their username/password combination and will grant access when the credentials match those stored on the TOE. The remote CLI is protected by SSH and allows users to authenticate against the TOE's local authentication mechanisms with either their username/password combination or SSH public key and will grant access when the credentials match those stored on the TOE.

The GUI can be accessed through a web browser and is protected by HTTPS/TLS. The only pre-authentication service that the TOE allows via the GUI is displaying the warning banner; which can be configured by a Security Administrator through the GUI. The GUI allows users to authenticate with their username/password combination against the TOE's local authentication mechanism and will grant access when the credentials match those stored on the TOE.

8.4.4 FIA_UAU.7:

When a user enters their password at the local CLI, the password characters entered by the Exabeam user are not echoed back to the local CLI.

8.4.5 FIA_X509_EXT.1/Rev, FIA_X509_EXT.1/ITT, FIA_X509_EXT.2, and FIA_X509_EXT.3:

The TOE uses X.509v3 certificates to support authentication for internal and external TLS communication. For internal communication between EX3000 and EX4000, EX3000 and EX4000 will verify their counterpart's certificate. For external, the EX3000 and the EX4000 will verify the syslog server's certificate.

The TSF determines the validity of certificates by ensuring that the certificate and the certificate path is valid in accordance with RFC 5280. In addition:

- The TOE components support a minimum path length of three certificates for syslog server certificates. While the TOE components support a minimum path length of two certificates for TOE component certificates.
- The certificate path must terminate with a trusted CA certificate.
- The TSF validates a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF validates the extendedKeyUsage field according to the following rules:

    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
- The TSF validates the revocation status of the certificate using OCSP in accordance with RFC 6960.

The TOE requests an OCSP Responder, which resides in the operating environment, to check the revocation status of certificates provided in a certificate chain. It is expected that the OCSP Responder has the same physical controls and security provided to the TOE. When the TSF cannot establish a connection to determine the validity of a certificate, the TSF will not accept the certificate and not establish the connection with the other TOE component or operational environment entity. The TSF does not provide a management mechanism to override the validation decision.

A Certificate Request containing the "Common Name" and public key is generated, as specified in RFC 2986, by the Exabeam user on the remote CLI of each TOE component. Once created, the Certificate Request can be manually transferred to the CA for signature and then manually transferred back to the TOE component. As part of loading the CA Certificate Response, the Exabeam user will execute a command for the TOE to validate the chain of certificates from the Root CA.

8.5 Security Management

With respect to TOE security functions as defined by the SFRs, each TOE component performs the same user authentication functions and management functionality except for the differences with inter-TOE communication where the Security Administrators manage the EX3000 as the client and EX4000 as the server within the distributed TOE.

8.5.1 FMT_MOF.1/ManualUpdate:

The TOE restricts the ability to perform manual updates to the Security Administrator (i.e. the Exabeam user) via the remote CLI on each TOE component. There are no other methods for updating the TOE.

8.5.2 FMT_MTD.1/CryptoKeys:

The ability to modify, delete, and generate/import cryptographic keys is limited to the Exabeam user through the CLI on each TOE component.

8.5.3 FMT_MTD.1/CoreData and FMT_SMR.2:

There are two types of user accounts on each TOE component, those that access the TOE through the CLI, and those that access through the GUI. The CLI can be accessed locally through a keyboard and monitor or remotely through an SSH session. The GUI can only be accessed remotely. The only administrative action allowed before authentication is the ability to view the security banner for the GUI and the CLI. All further management of the TOE and its TSF data is limited based upon the TOE's authentication mechanisms, the available user accounts on each interface, and the access control policies.

The Exabeam user provides the majority of the management of the TOE security functions and is the only Security Administrator for the remote CLI. For the local CLI, the Exabeam user is the primary Security Administrator but there is also the root account which can perform the entire set of security functions which are available to the Exabeam user. Thus, in all cases where the Security Target states the Exabeam user can perform a function, the root user can also perform that function. The root user can also perform additional audit management functions, unlock the Exabeam user account due to failed authentication attempts, configure the reference and/or expected identifiers on the TOE components, and cannot have its account become locked. The TOE's CLI access control policies differentiate these functions between the Exabeam user and root user roles. The Exabeam user has the ability to assume the role of root to perform management activities. It is recommended that the Exabeam user always be used over the root account for management.

The GUI has users, which can belong to one or more roles and the TOE enforces a role-based access control (RBAC) policy based upon the role(s) assigned to a user. Each role defines a set of permissions and the permissions that can perform TSF functionality are 'Manage Users and Context Sources' and 'Manage Context Tables'. The default role with these permissions is the 'Administrator' role but the TOE allows the definition of new roles with these permissions by a user with the 'Administrator' role. Therefore, any user with the 'Administrator' role or any role created with one or both of these permissions is considered a Security Administrator. The GUI also has a default user account called the 'Admin' user which belongs to the 'Administrator' role and is required to have its default password changed in the evaluated configuration. However, the 'Admin' user cannot change its name and the 'Administrator' role cannot have its permissions changed.

8.5.4 **FMT_SMF.1:**

The TOE has two types of users on each TOE component, those that access the TOE through the CLI, and those that access through the GUI. The CLI allows management of the TOE remotely and locally, while the GUI allows only remote management. The role of administrator for the CLI is fulfilled by the Exabeam user, while for the GUI it is fulfilled by either the 'Administrator' role or roles that contain the 'Manage Users and Context Sources' and/or 'Manage Context Tables' permissions. The Security Administrators are capable of performing the following management functions on the TOE as defined elsewhere in this document:

- Ability to administer the TOE locally (local CLI) and remotely (GUI and remote CLI);
- Ability to configure the GUI access banner via the GUI and the CLI access banner via the local or remote CLI;
- Ability to configure the session inactivity time before session termination via the local or remote CLI;
- Ability to update the TOE, and to verify the updates using a published hash prior to installing those updates via the remote CLI;
- Ability to configure the authentication failure parameters for FIA_AFL.1 via the local or remote CLI;
- Ability to configure the cryptographic functionality via the local or remote CLI;

- Ability to configure the interaction between TOE components via the local and remote CLI;
- Ability to re-enable a locked Administrator account via the local CLI, remote CLI, or GUI;
- Ability to set the time which is used for time-stamps via the local and remote CLI.

8.6 Protection of the TSF

8.6.1 FPT_ITT.1:

Once the TOE components have been configured to communicate, TLS is used for the transfer of collected network event data from EX3000 to EX4000. The connection is only established when collected network event data is ready to be sent and is terminated once all data has been transferred. If a connection cannot be established due to network failure, the EX3000 will continue attempting to establish a connection to the EX4000 until the connection can be established and does not require any administrative involvement.

8.6.2 FPT_APW_EXT.1:

There is no function provided by the TOE to display a password value in plaintext and both TOE components secure the passwords in the same manner. The local user store for the GUI credentials is a database which stores the passwords as a hashed value using SHA-256. The CLI user credentials are stored in /etc/passwd which hashes the password with SHA-512. The TOE does not provide GUI users access to user passwords and the Exabeam user is able to view the locations of these passwords but can only see their hashed values.

8.6.3 FPT_SKP_EXT.1:

The TOE does not contain any interface that was specifically designed to view any of its pre-shared keys, symmetric keys, and private keys, and both TOE components protect these keys in the same manner. The Diffie-Hellman Shared Secret, Diffie-Hellman Private Key, and SSH Session Keys are stored in volatile memory (RAM) and are not accessible by any user. Because this key data is stored in memory, core dumps are disabled to prevent this data from being disclosed if an error were to occur on the underlying operating system.

The SSH Server Host Private Key and TLS Server Host Certificate Private Key are stored on the local filesystem and RAM. When these keys are stored in RAM, they have the same protections as the other keys stored in volatile memory. After these keys are created by the Exabeam user via the local or remote CLI, they will be assigned permissions to prevent unauthorized access which is enforced by the TOE's CLI access controls. The Exabeam user also has the ability to delete SSH Server Host Private Key and TLS Server Host Certificate Private Key using the 'rm' command via the local or remote CLI.

8.6.4 FPT_STM_EXT.1:

Each TOE component has a software clock which is backed by an underlying hardware clock that is used for time keeping. The Exabeam user can set the time manually via the local or remote CLI. The TOE uses the clock for several security-relevant purposes, including:

- Audit records
- Inactivity timeout for local CLI sessions

- Inactivity timeout for remote CLI sessions
- Inactivity timeout for remote GUI sessions
- X509 certificate validation
- Locked out period for a locked account due to failed authentication attempts

8.6.5 **FPT_TST_EXT.1:**

Each TOE component performs its own Power-On Self Tests (POSTs) and the POSTs are the same for both TOE components. Upon the startup of a TOE component, all POSTs are executed, and additionally continuous conditional tests are performed while the TOE operates. Upon boot, each TOE component will check the integrity of its firmware and software images. The firmware and software images are hashed and the hash values are checked against a local registry of SHA-256 values for each firmware and software image. If the values match, the boot process continues to proceed. If at any time the mismatch occurs, the boot process stops, and the TOE component will enter an error state.

Additionally, the TOE's cryptographic module will test its integrity using an HMAC-SHA-256 whenever the device is restarted. The integrity test verifies that the module has not been compromised and ensures that the results of the entropy mechanism are reliable. When a self-test fails the cryptographic module will go into a hard error state. Further cryptographic operations are prevented until the error state is cleared; which will occur when the TOE component is powered off and then powered back on again, causing the cryptographic module to be reloaded.

The cryptographic module performs a DRBG Known Answer Test (KAT), where a calculated value is compared to a stored value to verify correct operation, together with a Continuous Random Number Generator Test (CRNGT), which compares the current generated value with the previous generated value. This test ensures consecutive random numbers do not repeat. If the DRBG does repeat numbers, it will restart. However, the DRBG will only produce the same output if it is given the same inputs twice which would require a statistical anomaly to occur based upon the calculated entropy rate defined in the proprietary Entropy Analysis Report provided to NIAP. In addition, DRBG health tests are performed as required by SP 800-90 section 11.

These self-tests are sufficient to validate the correct operation of the TSF because they verify that the TOE component's firmware and software images have been unmodified through integrity checks and the TOE component's cryptographic module is operating correctly. The POSTs prevent the TOE component's software from executing in an unpredictable or inconsistent manner.

8.6.6 **FPT_TUD_EXT.1:**

The GUI on each TOE component provides the Security Administrator a means to determine the currently executing version of the TOE. The Exabeam user will SCP push (over SSH) the software package from their management workstation to each TOE component and then will run the commands to update each TOE component's software individually. The Security Administrator is made aware of new updates to the TOE by Exabeam sending an email with a link to an Exabeam hosted FTP server to download the latest software. The Security Administrator can also access Exabeam's website to check for the latest software version as well as contact customer support to request the latest software version.

The new version of the TOE component's software is downloaded to the management workstation. The installer package is transferred via an SCP push to the TOE component by the Exabeam user via the

remote CLI management interface. The Security Administrator will receive via email the SHA-256 hash value for the entire TOE software installer package (i.e. both header script and payload). Once the installer package has been transferred to the TOE, the Exabeam user can manually verify the hash of the installer package by running the following command in the same directory as the installer package: sha256sum -c checksums.txt. The output of this command will be either OK or FAILED. If FAILED is received Exabeam user must abort the installation process. If OK, the Exabeam user then initiates the installation process by executing the installer package. The TOE component will then check a SHA-256 hash that has been included within the header script of the installer package. This SHA-256 value is automatically verified against the included payload during initial extraction of the installer package. If the hash value is missing or the comparison does not result in a match, the installation will abort immediately. If the comparison does match, the TOE component's software is then updated and becomes the active version of the software.

8.7 TOE Access

8.7.1 FTA_SSL_EXT.1/FTA_SSL.3:

The Exabeam user via either the local or remote CLI can configure the maximum inactivity time period for the local console and remote CLI. The value is set in the /etc/profile.d/autologout.sh which contains the TMOUT variable and it can be set to a value between 1 to 36000 seconds. The default value is 7200 seconds. When the maximum time period of inactivity is reached, the local or remote CLI session will be terminated.

The maximum inactivity time period for the GUI can be configured by the Exabeam user via either the local or remote CLI by setting the value for webcommon.silhouette.authenticator.cookieIdleTimeout in the /opt/exabeam/config/common/web/custom/application.conf file. The value can be set between 60 and 86,400 seconds. The default value is set to 7,200 seconds (two hours). When the maximum time period of inactivity is reached, the GUI session will be terminated.

8.7.2 FTA_SSL.4:

Any user accessing the TOE via the CLI or the GUI can terminate their own session. On the GUI, EX3000 has 'Logout' button and EX4000 has a 'Sign Out' button. For the local and remote CLI, the Exabeam user can terminate their own session by using the 'exit' command on both the EX3000 and EX4000.

8.7.3 FTA_TAB.1:

There are three possible administrative ways to log into each TOE model: local CLI, remote CLI, and remote GUI application. When logging in locally or remotely, the pre-authentication banner is displayed and is viewed prior to authentication. The CLI authentication banner is administratively configurable in the CLI by the Exabeam user. The GUI authentication banner is administratively configurable in the GUI by the Security Administrator.

8.8 Trusted Path/Channels

8.8.1 **FTP_ITC.1:**

Both the EX3000 and EX4000 TOE components connect as a client to a syslog server in the operational environment via their own trusted channel. The channels are logically distinct from each other and do not interfere with the operation of the other channels of communication. The connections to the syslog server by the EX3000 and EX4000 are for the transfer of the TOE's audit records and the connections are secured with TLS. The TLS conforms to FCS_TLSC_EXT.1 for these connections. These protocols are used to protect the data traversing the trusted channels from disclosure and/or modification.

8.8.2 **FTP_TRP.1/Admin:**

Remote administration is secured by using SSH and TLS protocols. The TOE supports four remote administrative connections:

- EX3000 is a SSH server for its remote CLI
- EX4000 is a SSH server for its remote CLI
- EX3000 is a TLS server (HTTPS) for its GUI
- EX4000 is a TLS server (HTTPS) for its GUI

A web browser initiates the HTTPS connection to the TOE component's GUI on behalf of the user for remote administration. The TOE component is acting as a TLS server and is conformant to FCS_TLSS_EXT.1 and FCS_HTTPS_EXT.1. A user can connect to the TOE component's remote CLI using an SSH client and connects to the remote CLI to perform remote administration. The TOE component's SSH server implementation is conformant to FCS_SSHS_EXT.1. These protocols are used to protect the data traversing these trusted paths from disclosure and/or modification.