Forescout eyeSight v9.1

Security Target

ST Version: 2.0
September 09, 2025

Forescout Technologies, Inc.
190 West Tasman Drive
San Jose, CA, USA 95134

Prepared By:

Booz | Allen | Hamilton

delivering results that endure

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.

1.2 ST Identification

ST Title: Forescout eyeSight v9.1 Security Target
ST Version: 2.0
ST Publication Date: September 09, 2025
ST Author: Booz Allen Hamilton

1.2.1 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.

1.2.2 Terminology

This section defines the terminology used throughout this ST. The terminology used throughout this ST is defined in Table 1 & 2. These tables are to be used by the reader as a quick reference guide for terminology definitions.

Term	Definition
Appliance	A Forescout eyeSight with the CounterACT (CT) license applied. The CT license is for basic device visibility features.
Centralized Enterprise Manager (CEM)	A Forescout eyeSight with the CEM license applied. The CEM license unlocks centralized management functionality for managing multiple eyeSight Appliances.

Term	Definition
Console or Console application	The Forescout Console is a separate GUI application, installed on an administrative workstation, used for creating NAC, firewall and IPS policies, generating reports, viewing and managing detection information, and managing Forescout eyeSight.
Endpoint	A Network Host discovered by the Forescout eyeSight, for example desktop, laptop, server, etc.
Local CLI	When the TOE's command line interface (CLI) is accessed locally with a physical connection to the TOE via the keyboard/video ports or a serial port and a terminal emulator that is compatible with serial communications is referred to as the local CLI. Note: The NDcPP utilizes the term Local Console in its terminology and this is the same as the Local CLI.
Plugins	Functionality enhancement modules that can be incorporated into the Forescout eyeSight. Plugins enable deeper inspection as well as broader control over network endpoints. Bundled plugins are pre-packaged with the Forescout eyeSight. Other plugins may be available from Forescout or from a third party. Only the syslog and Active Directory plugins are included in the scope of the evaluation as they provide functional support for the NDcPP defined SFRs.

Table 1: Customer Specific Terminology

Term	Definition
Authorized Administrator	The claimed Protection Profile defines an Authorized Administrator role that is authorized to manage the TOE and its data. For the TOE, this is considered to be any user with the "administrator" role.
Security Administrator	Synonymous with Authorized Administrator and System Administrator.
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.).

Table 2: CC Specific Terminology

1.2.3 Acronyms

The acronyms used throughout this ST are defined in Table 3. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
AD	Active Directory
CC	Common Criteria
CEM	Centralized Enterprise Manager
CLI	Command-line Interface
CPU	Central Processing Unit
CT	CounterACT
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IT	Information Technology

Acronym	Definition
LDAP	Lightweight Directory Access Protocol
NIAP	National Information Assurance Partnership
OS	Operating System
PP	Protection Profile
RU	Rack Unit
SAR	Security Assurance Requirement
SCP	Secure Copy Protocol
SFP	Security Function Policy
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSL	Secure Sockets Layer
SSH	Secure Shell
ST	Security Target
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
UI	User Interface

Table 3: Acronym Definition

1.2.4 Reference

[1] collaborative Protection Profile for Network Devices Version 3.0e [NDcPP3.0e]

[2] Functional Package for Secure Shell (SSH) Version 1.0 [SSH FP]

[3] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated April 2017, version 3.1, Revision 5, CCMB-2017-004-001

[4] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, version 3.1, Revision 5, CCMB-2017-004-002

[5] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated April 2017, version 3.1, Revision 5, CCMB-2017-004-003

[6] Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated April 2017, version 3.1, Revision 5, CCMB-2017-004-004

1.3 TOE Reference

The TOE is Forescout eyeSight operating on software version 9.1. Forescout eyeSight is a product family, which includes the following models: 4130, 5120-01, 5120-02, 5140-01, 5140-02, 5160-01, 5160-02, 6120, 6140, and 6160.

1.4 TOE Overview

The TOE is the Forescout eyeSight product operating on software version 9.1 and is referred to as the Forescout eyeSight or TOE from this point forward. The Forescout eyeSight is used to dynamically identify and evaluate network infrastructure, devices and applications connected to the network, and provide enforcement of Network Access Policy (NAC) and Enterprise Conformance Policies. Forescout's agentless technology discovers, classifies and assesses devices. The Forescout eyeSight interrogates the network

infrastructure to discover devices as they connect to the network. After discovering a device, the Forescout eyeSight uses a combination of passive and active methods to classify the device according to its type and ownership. Based on its classification, the Forescout eyeSight then assesses the device security posture and allows organizations to set policies that establish the specific behavior the device is allowed to have while connected to a network.

The Forescout Console application (aka Console) is a separately installed Windows executable which provides an administrator with a graphical user interface to manage the TOE. The Console must be installed on a separate Windows OS host platform. The Console communicates with the TOE via a secure TLS channel, shown as a yellow circle with a 3 in figure below. All external interfaces will be referred to as E#.

The TOE also provides a Command Line Interface (CLI) for remote and local management of the device. To access the CLI an administrator must either be locally connected (E1), via the keyboard/video or the serial port connections, or use SSHv2 to establish a secure connection (E2).

The CLI provides lower level configuration of the device such as initial IP address configuration which cannot be done via the Console, and some diagnostic capabilities. The CLI does not provide any OS-level or shell type access to the embedded OS on the TOE.

The following figure depicts the TOE boundary, operational environment, and external interfaces:

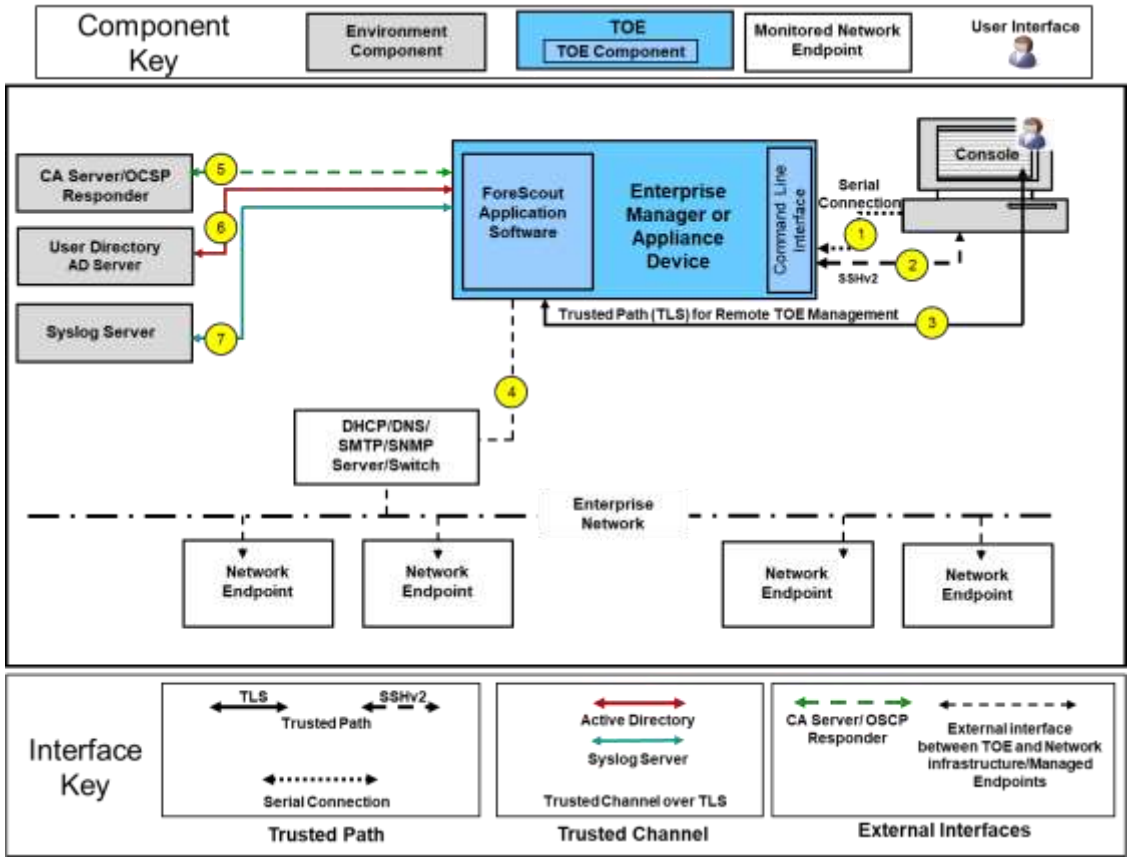

Figure 1: TOE Boundary

The Forescout device communicates with an audit server (E7), Active Directory Server (E6), Certificate Authority Server (E5), and the remote management workstation (with the Console) over a dedicated out-of-

band network management connection (E3). The connection to the enterprise network (E4) is a separate connection to the enterprise network environment that the TOE is monitoring and managing. A detailed description of each interface's operational environment component is in Table 5 below.

The Forescout eyeSight can be configured as a Centralized Enterprise Manager (CEM) or as an Appliance. The CEM configuration provides all of the functionality of an Appliance and provides an additional centralized hierarchical management functionality over Appliances. The additional functionality provided by the CEM is considered outside the scope of the evaluation because hierarchical management functions do not trace or map to any NDcPP3.0e requirements. The TOE, regardless of being configured as a CEM or Appliance, claims conformance to all NDcPP3.0e requirements as a standalone entity. Therefore, the TOE was tested as a standalone entity in both configurations to ensure that the claimed NDcPP3.0e functionality was conformant regardless of the TOE being configured as a CEM or an Appliance.

1.5 TOE Type

The TOE type for this product is a standalone network device that is used to dynamically identify and evaluate network infrastructure, devices and applications connected to the network, and provide enforcement of Network Access Policy (NAC) and Enterprise Conformance Policies.

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 following table describes the TOE components in the evaluated configuration:

TOE Components	Hardware Components	Software Version
Forescout eyesight product family	4130, 5120-01, 5120-02, 5140-01, 5140-02, 5160-01, 5160-02, 6120, 6140, and 6160	Forescout eyeSight v9.1

Table 4: TOE Models

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 are used in the operational environment for the TOE's evaluated configuration.

Component	Definition

Management Workstation	Any general-purpose computer that is used by the administrator to manage the TOE. For the TOE to be managed remotely the management workstation is required to have:
	• Non-dedicated machine:
	  o 2GB memory
	  o 1GB disk space
	• OS running:
	  o Windows 7/8/8.1/10/11
	  o Windows Server 2008 / 2008 R2 / 2012 / 2012 R2 / 2016 / 2019
	  o Linux RHEL/CentOS 7.9 / 8
	  o macOS 10.12 / 10.13 / 10.14 / 10.15 / 11
	  o SSHv2 client installed to access the TOE's CLI
	• Forescout Console application (Console) installed
	
	TCP communications from the Management Workstation to the TOE is secured using:
	• SSH for remote access to the remote CLI
	• TLS for remote access from the Console
	
	The TOE's CLI can also be accessed locally with a physical connection to the TOE using the keyboard/video or the serial port and must use a terminal emulator that is compatible with serial communications (local CLI).
	
	The TOE acts as a server for both protocols. This OE component is required to support interfaces E1, E2, & E3 as defined in Figure 1 above.

Active Directory Server	A system that is capable of receiving authentication requests over TLS and validating these requests against identity and credential data that is defined in the directory (Microsoft version of an LDAP Server). The TOE is the TLS client for this communication. Required to support interface E6 as defined in Figure 1 above.

Component	Definition
Audit Server	The TOE connects to an audit server to send the audit records 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. This OE component is required to support interface E7 as defined in Figure 1 above.
Certificate Authority (CA) Server/Online Certificate Status Protocol (OCSP) Responder	Certificate authority servers issue and revoke digital certificates. The OCSP responder (a server typically run by the certificate issuer) will, when queried for revocation status of a certificate chain, returns a signed response signifying that the certificate specified in the request is 'good', 'revoked', or 'unknown'. This OE component is required to support interface E5 as defined in Figure 1 above.
Network Infrastructure	The network infrastructure contains components such as routers, switches, DNS server, etc. Figure 1 identifies these interfaces as a single interface. The interface to the managed network infrastructure is a separate connection to the enterprise operational environment the product is managing. The TOEs management of the enterprise operational environment is out of scope for the NDcPP3.0e. Therefore, interface E4 to these components is out of scope of the evaluation.

Table 5: Supporting Components in the Operational Environment

2.3 Excluded from the TOE

The following product 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

There are no components, applications, and/or functionality that are not installed.

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

The product includes a number of functions as described in Section 1.4, that are outside the scope of the claimed Protection Profile because the functionality cannot be mapped to any NDcPP3.0e SFRs. The excluded functionality is described below:

- Ability to dynamically identify and evaluate network infrastructure, devices and applications connected to the network, and providing enforcement of Network Access Policy (NAC) and Enterprise Conformance Policies.
- Forescout's agentless technology that is used for interrogating the network infrastructure to discover, classify, assess devices, assign device a security posture base on organizational policies, and establish specific behavior the device is allowed to have while connected to the network.
- The centralized hierarchical management functionality of the Centralized Enterprise Manager configuration over Appliances.

The only included evaluated functionality is scoped to those security functions defined by the NDcPP3.0e and defined in Section 6 of this document.

2.4 Physical Boundary

The following table outlines the models and their key differentiators that are part of the evaluation.

Equipment			
Software/Firmware	**Hardware Model**	**Supported Software License**	**Component/Configuration**
Forescout eyeSight v9.1	4130	CT	1U Rack-mount
			1 HDD
			1 x Gen 8 Intel® Core™ i5-8500T (Coffee Lake)
			6 x Intel-based NIC Ethernet Ports
	5120-01	CT & CEM	1U Rack-mount
			3 HDD (RAID1+HS)
			1 x Xeon Silver 4110 (Skylake)
			4 (up to 8)x Intel-based NIC Ethernet Ports
	5120-02	CT & CEM	1U Rack-mount
			3 HDD (RAID1+HS)
			1 x Xeon Silver 4208 (Cascade Lake)
			4 (up to 8)x Intel-based NIC Ethernet Ports
	5140-01	CT & CEM	1U Rack-mount
			3 HDD (RAID1+HS)
			2 x Xeon Silver 4114 (Skylake)
			4 (up to 8)x Intel-based NIC Ethernet Ports
	5140-02	CT & CEM	1U Rack-mount
			3 HDD (RAID1+HS)
			2 x Xeon Silver 4210 (Cascade Lake)
			4 (up to 8)x Intel-based NIC Ethernet Ports
	5160-01	CT & CEM	1U Rack-mount
			3 HDD (RAID1+HS)
			2 x Xeon Gold 6132 (Skylake)
			4 (up to 8)x Intel-based NIC Ethernet Ports
	5160-02	CT & CEM	1U Rack-mount
			3 HDD (RAID1+HS)
			2 x Xeon Gold 6226R (Cascade Lake)
			4 (up to 8)x Intel-based NIC Ethernet Ports
	6120	CT & CEM	1U Rack-mount
			3 HDD (RAID1+HS)
			1 x Xeon Silver 4410Y (Saphire Rapids)
			1 Intel-based NIC Ethernet Port
	6140	CT & CEM	1U Rack-mount
			3 HDD (RAID1+HS)
			2 x Xeon Silver 4410Y (Saphire Rapids)
			1 Intel-based NIC Ethernet Port
	6160	CT & CEM	1U Rack-mount
			3 HDD (RAID1+HS)
			2 x Xeon Gold 5418Y (Saphire Rapids)
			1 Intel-based NIC Ethernet Port

Table 6: 4130, 51xx, and 61xx Models

2.5 Logical Boundary

The TOE is comprised of the following security features that have been scoped by the protection profile:

- Security Audit
- Cryptographic Support
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

2.5.1 Security Audit

The TOE contains mechanisms to generate audit data to record predefined events on the TOE. The generates application layer audit events and OS log files. Both are stored in the TOE's local hard drive. An administrator has the ability to configure the TOE to forward events to an audit server. In the evaluated configuration, the audit data is also securely transmitted to the audit server using a TLS v1.2 communication channel.

2.5.2 Cryptographic Support

The TOE provides cryptography in support of SSH and TLS (v1.2) trusted communications. Two different cryptography software packages are included with the TOE: Bouncy Castle and OpenSSL. Bouncy Castle is used specifically for communications with the management workstation running the Console. OpenSSL is used for all other TLS and SSH communications. The TOE immediately destroys keys when no longer used. The following table identifies the cryptographic services per cryptographic library.

SFR	OpenSSL Implementation CAVP #A7369	Bouncy Castle Implementation CAVP #A7362
FCS_CKM.1	RSA per FIPS 186-4 Key Generation	N/A
	ECC schemes using 'NIST curves' P-256, P-384, P-521, per FIPS PUB 186-4	N/A
FCS_CKM.2	RSA Key Establishment per RSAES-PKCS-v1_5	RSA Key Establishment per RSAES-PKCS-v1_5
	Elliptic curve-based Key Establishment NIST Special Publication 800-56A Revision 3	N/A
FCS_COP.1/DataEncryption	AES CTR 128 and 256 bits AES GCM 128 and 256 bits	AES CBC: 128 and 256 bits AES GCM: 256 bits
FCS_COP.1/SigGen	RSA FIPS 186-4 Signature Services 2048 bits	RSA FIPS 186-4 Signature Services 2048 bits
FCS_COP.1/Hash	SHA-256 SHA-384 SHA-512	SHA-1 SHA-256 SHA-384 SHA-512
FCS_COP.1/KeyedHash	HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-384

FCS_RBG_EXT.1	CTR DRBG (AES-256)	Hash DRBG [SHA-256, SHA-384, SHA-512]

Table 7: Cryptographic Services

2.5.3 Identification and Authentication

The TSF provides a configurable number of maximum consecutive authentication failures that are permitted by a user. Once this number has been met, the account is locked for a configurable time interval or until a Security Administrator manually unlocks the account.

The TOE provides local password authentication for CLI and Console users as well as providing the ability to securely connect to an Active Directory server for the authentication of Console users. Communications over this interface is secured using TLS in which the TOE is acting as a client. The TOE enforces the use of X.509 certificates to support authentication for TLS connections. The only function available to an unauthenticated user is the ability to acknowledge a warning banner. Passwords that are maintained by the TSF can be composed of upper case, lower case, numbers and special characters. A Security Administrator can define the minimum password length between 6 and 100 characters.

2.5.4 Security Management

The TOE can be administered either locally or remotely. Role-based access control is used to prevent unauthorized management and access to TSF data.

2.5.5 Protection of the TSF

The TOE ensures the security and integrity of all data that is stored locally and accessed remotely. Passwords are not stored in plaintext. A administrator has the ability to query the TOE for the currently executing version the TOE software and is required to manually initiate the update process from the Console as automatic updates are not supported. The TOE automatically verifies the digital signature of the software update prior to installation. If the digital signature is found to be invalid, the administrator is provided with an error and the update is not installed. There is no means for an administrative override to continue the installation if the signature is completely missing. The TOE implements a self-testing mechanism that is automatically executed during the initial start-up and can be manually initiated by an administrator after authentication. The TOE's self-tests verify the correct operation of product and cryptographic modules. The TOE provides its own time via its internal clock.

2.5.6 TOE Access

The TOE displays a configurable warning banner prior to its use. Inactive sessions will be terminated after an administratively-configurable time period. Users are allowed to terminate their own interactive session. Once a remote session has been terminated, the TOE requires the user to re-authenticate to establish a new session. Local and remote sessions are terminated after the administratively-configured inactivity time limit is reached.

2.5.7 Trusted Path/Channels

Users can access a CLI for administration functions remotely via SSH (remote CLI) or a local physical connection (local CLI) to the TOE. The TOE operates as an SSH server for the remote CLI. The Console is the main administrator interface, which is running on a separate Windows PC and requires the use of TLS to communicate with the TOE. The TOE operates as a TLS server for requests from the Console.

The TOE acts as a TLS client to initiate the following secure paths for the following functions to their associated operational environment entities:

- User authentication (Active Directory)
- Auditing (audit server)

3 Conformance Claims

3.1 CC Version

This ST is compliant with Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 5 April 2017.

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 September 09, 2025.

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 September 09, 2025.

3.4 PP Claims

This ST claims exact conformance to the Collaborative Protection Profile for Network Devices Version 3.0e (NDcPP3.0e), December 6, 2023.

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

- FCS_TLSC_EXT.1
- FCS_TLSS_EXT.1
- FIA_X509_EXT.1/Rev
- FIA_X509_EXT.2
- FIA_X509_EXT.3
- FIA_AFL.1
- FIA_UAU.7
- FIA_PMG_EXT.1
- FPT_APW_EXT.1
- FMT_MTD.1/CryptoKeys
- FTA_SSL_EXT.1

The TOE does not claim any Optional SFRs that are defined in the appendices of the claimed PPs.

This does not violate the notion of exact conformance because the PPs specifically indicate 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.

3.5 Package Claims

Additionally, the TOE claims exact compliance to the Functional Package for Secure Shell Version 1.0, (SSH FP), May 13, 2021. The TOE claims following Selection-Based SFRs that are defined in the claimed PP:

- FCS_SSHS_EXT.1

3.6 Package Name Conformant or Package Name Augmented

This ST and TOE are in exact conformance with the SSH FP which is conformant with CC Part 3 to include all applicable NIAP and International interpretations through June 23, 2025.

3.7 Technical Decisions

Technical Decisions that effected the SFR wording have been annotated with a Footnote.

The following list of the NDcPP3.0e and SSH FP Technical Decisions apply to the TOE because SFR wording, application notes, or assurance activities were modified for SFRs claimed by the TOE:

Table 8: Technical Decisions

TD #	Title	References	Changes			Analysis to this evaluation	
			SFR	AA	Notes	NA	Reason
TD0836	Redundant Requirements in FPT_TST_EXT.1	FPT_TST_EXT.1 [NDcPP3.0e]	X	X	X		Update to SFR, application notes, and Assurances activities Footnote 5
TD0868	Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	FCS_IPSEC_EXT.1.7, FCS_IPSEC_EXT.1.8 [NDcPP3.0e]	X		X	X	Updates to SFR and Application Notes IPSEC not claimed
TD0879	Correction of Chapter Headings in CPP_ND_V3.0E	Appendix B.7 [NDcPP3.0e]			X		Updates to Appendix B
TD0880	Removal of Duplicate Selection in FMT_SMF.1.1	FMT_SMF.1.1 [NDcPP3.0e]	X		X		Updates to SFR and Application Notes Footnote 4
TD0886	Clarification to FAU_STG_EXT.1 Test 6	FAU_STG_EXT.1 [NDcPP3.0e]			X		Updates to the Application notes in the SD
TD0899	NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2	FCS_TLSC_EXT.1.9, FCS_TLSS_EXT.1.8, [NDcPP3.0e-SD]		X			Updates to testing assurance activities
TD0900	NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	FIA_UIA_EXT.1.3 [NDcPP3.0e]	X		X		Updates to the SFR and Application Notes Footnote 3
TD0921	NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment	FCS_CKM.1, FCS_COP.1/SigGen, CPP_ND_V3.0e-SD [NDcPP3.0e]	X	X			Updates to the SFR Updates to and test Assurances activities Footnote 1 & 2
TD0923	NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2	FAU_GEN.1.2 [NDcPP3.0e]			X		Updates and application note

3.8 Conformance Claim Rationale

Section 1.2 of the NDcPP3.0e states: The NDcPP defines a network device as "a device that is connected to a network and has an infrastructure role within that network. The TOE may be standalone or distributed, where a distributed TOE is one that requires multiple distinct components to operate as a logical whole in order to fulfil the requirements of this cPP…" Additionally, the NDcPP3.0e says that example devices that fit this definition include "physical and virtualised routers, firewalls, VPN gateways, IDSs, and switches."

The TOE is a standalone network device, composed of hardware and software, that is connected to the network and enables network access control, threat protection, and compliance of the entire enterprise based on network security policies. Therefore, the TOE provides an infrastructure role in internetworking of different network environments across an enterprise.

The Forescout eyeSight product are devices that are used to dynamically identify and evaluate network infrastructure, devices and applications connected to the network, and to provide enforcement of Network Access Policy (NAC) and Enterprise Conformance Policies. Based on the TOE being a network device that provides an infrastructure role within a network, the TOE product type classification is justified and the NDcPP conformance claim is appropriate.

4 Security Problem Definition

4.1 Threats

This section identifies the threats against the TOE. These threats have been taken from the NDcPP3.0e.

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

Threat	Threat Definition
	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. Threat agents may also be able to take advantage of weak administrative passwords to gain privileged access to the device.
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 9: TOE 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 NDcPP3.0e.

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 Administrators consent by accessing the TOE.

Table 10: TOE Organization Security Policies

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 NDcPP3.0e.

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 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 does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does 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. For vNDs, this assumption applies to the physical platform on which the VM runs.
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). If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside

Assumption	Assumption Definition
	more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform.
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 ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules 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 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. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
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.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 11: TOE 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.

4.4.1 TOE Security Objectives

The NDcPP3.0e does not define any security objectives for the TOE.

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

Objective	Objective Definition
	necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.
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. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
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.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. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

Table 12: 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 PPs to which the ST and TOE claim conformance. These extended components are formally defined in the PPs in which their usage is required.

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. "/LocSpace" for an SFR that relates to local storage space

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 PPs (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
Cryptographic Support	FCS_CKM.1	Cryptographic Key Generation
	FCS_CKM.2	Cryptographic Key Establishment
	FCS_CKM.4	Cryptographic Key Destruction
	FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
	FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
	FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
	FCS_RBG_EXT.1	Random Bit Generation
	FCS_SSH_EXT.1	SSH Protocol
	FCS_SSHS_EXT.1	SSH Server Protocol
	FCS_TLSC_EXT.1	TLS Client Protocol
	FCS_TLSS_EXT.1	TLS Server Protocol
Identification and Authentication	FIA_AFL.1	Authentication Failure Handling
	FIA_PMG_EXT.1	Password Management
	FIA_UAU.7	Protected Authentication Feedback
	FIA_UIA_EXT.1	User Identification and Authentication
	FIA_X509_EXT.1/Rev	X.509 Certificate Validation
	FIA_X509_EXT.2	X509 Certificate Authentication
	FIA_X509_EXT.3	X509 Certificate Requests

Class Name	Component Identification	Component Name
Security Management	FMT_MOF.1/ManualUpdate	Management of Security Functions Behavior
	FMT_MTD.1/CoreData	Management of TSF Data
	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_APW_EXT.1	Protection of Administrator Passwords
	FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric 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 13: 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 Administrator account shall be logged if individual 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 Administrator account shall be logged)];

d) Specifically defined auditable events listed in **Table 14**.

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 14**.

Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_SSH_EXT.1	[Failure to establish SSH connection].	Reason for failure. [Non-TOE endpoint of attempted connection (IP Address)].
	[Establishment of SSH connection].	[Non-TOE endpoint of connection (IP Address)].
	[Termination of SSH connection session].	[Non-TOE endpoint of connection (IP Address)].
	[None].	[None].
FCS_TLSC_EXT.1	Failure to establish a TLS session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded	Origin of the attempt (e.g., IP address)
FIA_PMG_EXT.1	None.	None.
FIA_UAU.7	None.	None.
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TST_EXT.1	None.	None.

Requirement	Auditable Event(s)	Additional Audit Record Contents
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FTA_SSL_EXT.1 (if "terminate the session" is selected)	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.
FTA_TAB.1	None.	None.
FTP_ITC.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	• None • None • Reason for failure
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failures of the trusted path functions.	• None • None • Reason for failure

Table 14: 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. In addition [

- The TOE shall consist of a single standalone component that stores audit data locally,

].

FAU_STG_EXT.1.3

The TSF shall maintain a [log file, database] of audit records in the event that an interruption of communication with the remote audit server occurs.

FAU_STG_EXT.1.4

The TSF shall be able to store [persistent] audit records locally with a minimum storage size of [*75 MB for OS log files and 30 MB for database audit records*].

FAU_STG_EXT.1.5

The TSF shall [overwrite previous audit records according to the following rule: [*rotate OS log files by deleting the oldest OS log file, rolling over the remaining OS log files, and create a new OS log file for the new audit records; purge database audit records by deleting the oldest entries based on first-in-*

first-out (FIFO) rule and generating an audit record for the purge event]] when the local storage space for audit data is full.

FAU_STG_EXT.1.6

The TSF shall provide the following mechanisms for administrative access to locally stored audit records [ability to view locally].

6.3.2 Class FCS: Cryptographic Support

6.3.2.1 *FCS_CKM.1 Cryptographic Key Generation*

FCS_CKM.1.1[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*] that meet the following: FIPS PUB 186-4, Digital Signature Standard (DSS), Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;
- ECC schemes using 'NIST curves' [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, Digital Signature Standard (DSS), Appendix B.4, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.;

].

6.3.2.2 *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 2.2";
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography;

].

6.3.2.3 *FCS_CKM.4 Cryptographic Key Destruction*

FCS_CKM.4.1

[1] TD0921

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite consisting of [zeroes]];
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [
    - logically addresses the storage location of the key and performs a [single-pass] overwrite consisting of [zeroes];
    ]

that meets the following: No Standard.

6.3.2.4 *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].

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

FCS_COP.1.1/SigGen[2]

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 [

- For RSA: [*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 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,

].

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

FCS_COP.1.1/Hash

[2] TD0921

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.

6.3.2.7 *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-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, implicit] and cryptographic key sizes [*160, 256, 384, 512*] and message digest sizes [160, 256, 384, 512] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 MAC Algorithm 2.

6.3.2.8 *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 [Hash_DRBG [SHA-256, SHA-384, SHA-512], CTR_DRBG (AES)].

FCS_RBG_EXT.1.2

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

6.3.2.9 *FCS_SSH_EXT.1 SSH Protocol*

FCS_SSH_EXT.1.1

The TOE shall implement SSH acting as a [server] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [4344, 5647, 5656, 6668] and no other standard.

FCS_SSH_EXT.1.2

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [

- "password" (RFC 4252),
- "publickey" (RFC 4252): [
  - rsa-sha2-256 (RFC 8332),

]

] and no other methods.

FCS_SSH_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [*262,130 bytes*] in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4

The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [

- aes128-ctr (RFC 4344),
- aes256-ctr (RFC 4344),
- aes128-gcm@openssh.com (RFC 5647),
- aes256-gcm@openssh.com (RFC 5647)

] and no other mechanisms.

FCS_SSH_EXT.1.5

The TSF shall protect data in transit from modification, deletion, and insertion using: [

- hmac-sha2-256 (RFC 6668),
- hmac-sha2-512 (RFC 6668),
- implicit

] and no other mechanisms.

FCS_SSH_EXT.1.6

The TSF shall establish a shared secret with its peer using: [

- ecdh-sha2-nistp256 (RFC 5656),
- ecdh-sha2-nistp384 (RFC 5656),
- ecdh-sha2-nistp521 (RFC 5656),

] and no other mechanisms.

FCS_SSH_EXT.1.7

The TSF shall use SSH KDF as defined in [

- RFC 5656 (Section 4)

] to derive the following cryptographic keys from a shared secret: session keys.

FCS_SSH_EXT.1.8

The TSF shall ensure that [

- a rekey of the session keys,

] occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

6.3.2.10 *FCS_SSHS_EXT.1 SSH Server Protocol*

FCS_SSHS_EXT.1.1

The TSF shall authenticate itself to its peer (SSH Client) using: [

- rsa-sha2-256 (RFC 8332)].

6.3.2.11 *FCS_TLSC_EXT.1 TLS Client Protocol Without Mutual Authentication*

FCS_TLSC_EXT.1.1

The TSF shall implement [TLS 1.2 (RFC 5246)] supporting the following ciphersuites: [

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

] and no other ciphersuites.

FCS_TLSC_EXT.1.2

The TSF shall verify that the presented identifier matches [the reference identifier per RFC 6125 Section 6 and no other attribute types].

FCS_TLSC_EXT.1.3

The TSF shall not establish a trusted channel if the server certificate is invalid [

- without any administrator override mechanism.

].

FCS_TLSC_EXT.1.4

The TSF shall [present the Supported Groups Extension with the following

curves/groups: [secp256r1, secp384r1, secp521r1]] in the Client Hello.

FCS_TLSC_EXT.1.5

The TSF shall [

- present the signature_algorithms extension with support for the following algorithms: [
  - rsa_pkcs1 with sha256(0x0401),
  - rsa_pkcs1with sha384(0x0501),
  - rsa_pkcs1 with sha512(0x0601),

  ] and no other algorithms;

].

FCS_TLSC_EXT.1.6

The TSF [does not provide] the ability to configure the list of supported ciphersuites as defined in FCS_TLSC_EXT.1.1.

FCS_TLSC_EXT.1.7

The TSF shall prohibit the use of the following extensions:

- Early data extension
- Post-handshake client authentication according to RFC 8446, Section 4.2.6.

FCS_TLSC_EXT.1.8

The TSF [not use PSKs].

FCS_TLSC_EXT.1.9

The TSF shall [reject [TLS 1.2] renegotiation attempts].

6.3.2.12 *FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication*

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_256_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288

] and no other ciphersuites.

FCS_TLSS_EXT.1.2

The TSF shall authenticate itself using X.509 certificate(s) using [RSA with key size [2048] bits and no other curves].

FCS_TLSS_EXT.1.3

The TSF shall perform key exchange using: [

- RSA key establishment with key size [2048] bits;

].

FCS_TLSS_EXT.1.4

The TSF shall support [session resumption based on session IDs according to RFC 5246 (TLS 1.2)].

FCS_TLSS_EXT.1.5

The TSF [does not provide] the ability to configure the list of supported ciphersuites as defined in FCS_TLSS_EXT.1.1.

FCS_TLSS_EXT.1.6

The TSF shall prohibit the use of the following extensions:

- Early data extension.

FCS_TLSS_EXT.1.7

The TSF shall [not use PSKs].

FCS_TLSS_EXT.1.8

The TSF shall [reject [TLS 1.2] renegotiation attempts].

6.3.3 Class FIA: Identification and Authentication

6.3.3.1 *FIA_AFL.1 Authentication Failure Management*

FIA_AFL.1.1

The TSF shall detect when an Administrator configurable positive integer within [*1-10*] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

FIA_AFL.1.2

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

6.3.3.2 *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 between [*6*] and [*100*] characters.

6.3.3.3 *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.

6.3.3.4 *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.

FIA_UIA_EXT.1.3[3]

The TSF shall provide the following remote authentication mechanisms [SSH password, SSH public key] and [external authentication server]. The TSF shall provide the following local authentication mechanisms [password-based].

FIA_UIA_EXT.1.4

The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

6.3.3.5 *FIA_X509_EXT.1/Rev X.509 Certificate Validation*

FIA_X509_EXT.1.1/Rev

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- 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.
- 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:
    - 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 DTLS/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 DTLS/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.

6.3.3.6 *FIA_X509_EXT.2 X.509 Certificate Authentication*

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].

[3]TD0900

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].

6.3.3.7 *FIA_X509_EXT.3 X.509 Certificate Requests*

FIA_X509_EXT.3.1

The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [Common Name, Organization, Organizational Unit, Country].

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 Class FMT: Security Management

6.3.4.1 *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.

6.3.4.2 *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 Security Administrators.

6.3.4.3 *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 Security Administrators.

6.3.4.4 *FMT_SMF.1 Specification of Management Functions*

FMT_SMF.1.1[4]

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

- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- [

[4] TD0880

- Ability to re-enable an Administrator account;
- Ability to set the time which is used for time-stamps;
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
- Ability to generate Certificate Signing Request (CSR) and process CA certificate response;
- Ability to administer the TOE locally;
- Ability to configure the local session inactivity time before session termination or locking;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to manage the trusted public keys database;

].

6.3.4.5 *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 remotely are satisfied.

6.3.5 Class FPT: Protection of the TSF

6.3.5.1 *FPT_APW_EXT.1 Protection of Administrator Passwords*

FPT_APW_EXT.1.1

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

FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext administrative passwords.

6.3.5.2 *FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)*

FPT_SKP_EXT.1.1

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

6.3.5.3 *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].

6.3.5.4 *FPT_TST_EXT.1 TSF Testing*

FPT_TST_EXT.1.1[5]

The TSF shall run a suite of the following self-tests:

- During initial start-up (on power on) to verify the integrity of the TOE firmware and software;
- Prior to providing any cryptographic services and [on-demand] to verify correct operation of cryptographic implementation necessary to fulfil the TSF;
- [no other] self-tests [*none*].

to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2

The TSF shall respond to [[*Hard-fail, Soft-fail*]] by [entering a maintenance mode, [*shutdown*]].

6.3.5.5 *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 [digital signature] prior to installing those updates.

6.3.6 Class FTA: TOE Access

6.3.6.1 *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.

6.3.6.2 *FTA_SSL.3 TSF-initiated Termination*

FTA_SSL.3.1

[5] TD0836

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

6.3.6.3 *FTA_SSL.4 User-initiated Termination*

FTA_SSL.4.1

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

6.3.6.4 *FTA_TAB.1 Default TOE Access Banner*

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.

6.3.7 Class FTP: Trusted Path/Channels

6.3.7.1 *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, [authentication server] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2

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

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for [*export audit, authentication decision*].

6.3.7.2 *FTP_TRP.1/Admin Trusted Path*

FTP_TRP.1.1/Admin

The TSF shall be capable of using [SSH, TLS] to 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 provides 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 PPs against which exact conformance is claimed and a subset of the 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.

Security Target (ASE)	ST introduction (ASE_INT.1)
	Conformance claims (ASE_CCL.1)
	Security Problem Definition (ASE_SPD.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Extended components definition (ASE_ECD.1)
	Stated security requirements (ASE_REQ.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance Documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life Cycle Support (ALC)	Labelling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability Assessment (AVA)	Vulnerability survey (AVA_VAN.1)

7.1 Class ASE: Security Target evaluation

7.1.1 ST introduction (ASE_INT.1)

7.1.1.1 *Developer action elements:*

ASE_INT.1.1D

The developer shall provide an ST introduction.

7.1.1.2 *Content and presentation elements:*

ASE_INT.1.1C

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

ASE_INT.1.2C

The ST reference shall uniquely identify the ST.

ASE_INT.1.3C

The TOE reference shall uniquely identify the TOE.

ASE_INT.1.4C

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

ASE_INT.1.5C

The TOE overview shall identify the TOE type.

ASE_INT.1.6C

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

ASE_INT.1.7C

The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C

The TOE description shall describe the logical scope of the TOE.

7.1.1.3 *Evaluator action elements:*

ASE_INT.1.1E

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

ASE_INT.1.2E

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

7.1.2 Conformance claims (ASE_CCL.1)

7.1.2.1 *Developer action elements:*

ASE_CCL.1.1D

The developer shall provide a conformance claim.

ASE_CCL.1.2D

The developer shall provide a conformance claim rationale

7.1.2.2 *Content and presentation elements:*

ASE_CCL.1.1C

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

ASE_CCL.1.2C

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

ASE_CCL.1.3C

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

ASE_CCL.1.4C

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

ASE_CCL.1.5C

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

ASE_CCL.1.6C

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

ASE_CCL.1.7C

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

ASE_CCL.1.8C

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

ASE_CCL.1.9C

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

ASE_CCL.1.10C

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

7.1.2.3 *Evaluator action elements:*

ASE_CCL.1.1E

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

7.1.3 Security problem definition (ASE_SPD)

7.1.3.1 *Developer action elements:*

ASE_SPD.1.1D

The developer shall provide a security problem definition.

7.1.3.2 *Content and presentation elements:*

ASE_SPD.1.1C

The security problem definition shall describe the threats.

ASE_SPD.1.2C

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

ASE_SPD.1.3C

The security problem definition shall describe the OSPs.

ASE_SPD.1.4C

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

7.1.3.3 *Evaluator action elements:*

ASE_SPD.1.1E

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

7.1.4 Security objectives for the operational environment (ASE_OBJ.1)

7.1.4.1 *Developer action elements:*

ASE_OBJ.1.1D

The developer shall provide a statement of security objectives.

7.1.4.2 *Content and presentation elements:*

ASE_OBJ.1.1C

The statement of security objectives shall describe the security objectives for the operational environment.

7.1.4.3 *Evaluator action elements:*

ASE_OBJ.1.1E

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

7.1.5 Extended components definition (ASE_ECD.1)

7.1.5.1 *Developer action elements:*

ASE_ECD.1.1D

The developer shall provide a statement of security requirements.

ASE_ECD.1.2D

The developer shall provide an extended components definition.

7.1.5.2 *Content and presentation elements:*

ASE_ECD.1.1C

The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C

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

ASE_ECD.1.3C

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

ASE_ECD.1.4C

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

ASE_ECD.1.5C

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

7.1.5.3 *Evaluator action elements:*

ASE_ECD.1.1E

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

ASE_ECD.1.2E

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

7.1.6 Stated security requirements (ASE_REQ.1)

7.1.6.1 *Developer action elements:*

ASE_REQ.1.1D

The developer shall provide a statement of security requirements.

ASE_REQ.1.2D

The developer shall provide a security requirements rationale.

7.1.6.2 *Content and presentation elements:*

ASE_REQ.1.1C

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

ASE_REQ.1.2C

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

ASE_REQ.1.3C

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

ASE_REQ.1.4C

All operations shall be performed correctly.

ASE_REQ.1.5C

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

ASE_REQ.1.6C

The statement of security requirements shall be internally consistent.

7.1.6.3 *Evaluator action elements:*

ASE_REQ.1.1E

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

7.1.7 TOE summary specification (ASE_TSS.1)

7.1.7.1 *Developer action elements:*

ASE_TSS.1.1D

The developer shall provide a TOE summary specification.

7.1.7.2 *Content and presentation elements:*

ASE_TSS.1.1C

The TOE summary specification shall describe how the TOE meets each SFR. In the case of entropy analysis, the TSS is used in conjunction with required supplementary information on Entropy.

7.1.7.3 *Evaluator action elements:*

ASE_TSS.1.1E

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

ASE_TSS.1.2E

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

7.2 Class ADV: Development

7.2.1 Basic Functional Specification (ADV_FSP.1)

7.2.1.1 *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.

7.2.1.2 *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.

7.2.1.3 *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.

7.3 Class AGD: Guidance Documentation

7.3.1 Operational User Guidance (AGD_OPE.1)

7.3.1.1 *Developer action elements:*

AGD_OPE.1.1D

The developer shall provide operational user guidance.

7.3.1.2 *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.

7.3.1.3 *Evaluator action elements:*

AGD_OPE.1.1E

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

7.3.2 Preparative Procedures (AGD_PRE.1)

7.3.2.1 *Developer action elements:*

AGD_PRE.1.1D

The developer shall provide the TOE including its preparative procedures.

7.3.2.2 *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.

7.3.2.3 *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.

7.4 Class ALC: Life Cycle Support

7.4.1 Labeling of the TOE (ALC_CMC.1)

7.4.1.1 *Developer action elements:*

ALC_CMC.1.1D

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

7.4.1.2 *Content and presentation elements:*

ALC_CMC.1.1C

The TOE shall be labeled with its unique reference.

7.4.1.3 *Evaluator action elements:*

ALC_CMC.1.1E

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

7.4.2 TOE CM Coverage (ALC_CMS.1)

7.4.2.1 *Developer action elements:*

ALC_CMS.1.1D

The developer shall provide a configuration list for the TOE.

7.4.2.2 *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.

7.4.2.3 *Evaluator action elements:*

ALC_CMS.1.1E

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

7.5 Class ATE: Tests

7.5.1 Independent Testing - Conformance (ATE_IND.1)

7.5.1.1 *Developer action elements:*

ATE_IND.1.1D

The developer shall provide the TOE for testing.

7.5.1.2 *Content and presentation elements:*

ATE_IND.1.1C

The TOE shall be suitable for testing.

7.5.1.3 *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.

7.6 Class AVA: Vulnerability Assessment

7.6.1 Vulnerability Survey (AVA_VAN.1)

7.6.1.1 *Developer action elements:*

AVA_VAN.1.1D

The developer shall provide the TOE for testing.

7.6.1.2 *Content and presentation elements:*

AVA_VAN.1.1C

The TOE shall be suitable for testing.

7.6.1.3 *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, Cryptographic Support, Identification and Authentication, Security Management, Protection of the TSF, TOE Access and Trusted Path/Channels.

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 TOE generates audit records for all administrative functions including Login/Logout, security related changes, resetting of passwords, and certificate management. Additionally, Table 14 identifies the audit records that are inclusive to the PPs evaluation scoping. The TOE records the date and time, type of event, subject identity (identity of the user associated with each audited event that occurred due to a user action), and the outcome in the audit record. The TOE associates each auditable event with the identity of the user that caused the event. For a full list of the audit events samples that are generated by the TOE, please refer to the Supplemental Administrative Guidance Document (AGD).

The TOE application layer maintains two separate log files in an internal database to record all the records needed to satisfy this requirement as scoped by the PP. The host OS also maintains an audit log (OS log) that is stored locally on the hard drive. All OS log records are incorporated into the appropriate application layer logs based on the type of event. The two application layer logs are as follows:

- **User Audit Trail**
  The User Audit Trail records information concerning TOE user activity for both CLI (OS log) and Console interface, for example: administrative changes to the security configuration of the TOE or updated/resetting of user passwords. The logs give additional information about the activity, such as the date of the activity and the IP address from which it was carried out.
- **System Event Log**
  The System Event Log records information about system activity, for example: successful and failed administrator authentication attempts, startup and shutdown of TOE or services, cryptographic key generation and destruction, and OS events. The startup and shutdown of the TOE's audit functionality is synonymous with the startup and shutdown of the TOE.

  The following is an example audit record for the Generating/import of, changing, or deleting of cryptographic keys (Timestamp, User: Admin, Event: Change Configuration with details: Fingerprint of certificate, Issued to, Issued by, Purpose/Use of certificate).

Generating/import of, changing, or deleting of cryptographic keys	2025-07-16T17:43:51-04:00 5140-Appliance 5140[2701313]: User admin session 8074980855193822285 changed Configuration. Details: Change trusted certificates configuration definition to <EOL> <EOL> Added: FingerPrint 'e50e2653e69a4bebf97dbc5d866ec48c4ccdbbb6' Issued to 'RootCA' Issued by 'RootCA'<EOL> Status enabled;Trusted For All subsystems on All devices;<EOL> Removed: FingerPrint 'e50e2653e69a4bebf97dbc5d866ec48c4ccdbbb6' Issued to 'RootCA'

	Issued by 'RootCA'<EOL> Status ena-bled;Trusted For All subsystems on All devices;<EOL> Changed: FingerPrint '6acadc4703a1fe1be5716a3d0c197a84b7c89d20' Issued to 'int01CA' Issued by 'rootCA'<EOL> Devices All=>All;Subsystems All=>All;<EOL> Changed: FingerPrint '8b291d227692dc67d4dad41c373a88b27b86acf2' Issued to 'rootCA' Issued by 'rootCA'<EOL> Devices All=>All;Subsystems All=>All;

8.1.2 FAU_STG_EXT.1

The TSF provides the ability for an administrator to configure the TOE for near real-time forwarding of the audit trail to an external audit server in the operational environment. The forwarding of the audit trail to an audit server is mandated for compliance to the NDcPP. Once configured and enabled, the generated audit is first saved locally in the internal database and then the TOE will securely transmit audit data to the Operational Environment audit server without administrator intervention via a TLS channel. During a connection outage to the audit server, the TOE continues to save audit locally. Once the connection to the audit server is re-established, the TOE automatically starts forwarding new audit records. The TOE does not forward the records created during the outage. Locally stored audit records are considered persistent. This is a standalone TOE that is responsible for storing and sending its own generated audit records.

Application layer audit events are stored in the TOE database (DB). The TSF runs an automatic DB purge function to prevent the application layer audit logs from filling up the internal database and hard drive to capacity. The DB, as part of the installation, determines a maximum size based on hard drive availability (approximately 30MB). This predefined and configurable threshold is used to trigger the DB purge function. The DB purge function is initiated when 75% of this predefined and non-configurable threshold is exceeded. When the DB threshold is exceeded, the DB purge function deletes entries in a “first-in, first-out” (oldest events deleted first) fashion. The DB purge function causes a syslog event to be sent by the TOE.

The TOE enforces a maximum size for the OS log file and the number of OS log files (current plus historical) saved at the OS level in order to control the amount of storage space needed for storing OS logs. For example, the TOE uses a maximum current OS log file size of 50MB and maintain 4 OS historical logs (1 current and 4 historical). The archived log files are approximately 6MB a piece. Therefore, the maximum total storage space required for the OS log files is calculated as 1*50MB + 4*6.1MB = 74.5 MB or approximately 75MB for the OS log files. Once the number of stored log files has reached the configured maximum number of maintained files, the oldest log file is automatically deleted and the remaining log files are rolled over in order to allow for the new file to be created for the new audit records. This first-in-first-out (FIFO) functionality protects the filesystem from becoming full and keeps the TSF within a defined maximum storage space.

The TOE provides a means to review all of the audit records via the Console interface. The TOE does not provide a means for any user to manually delete or manipulate the audit logs stored at the OS level or those in the internal DB. The management interfaces (Console or CLI) do not allow the audit records to be modified or deleted. The audit functionality starts automatically with the TOE and cannot be disabled by any means.

8.2 Cryptographic Support

The TOE implements two different cryptographic libraries: OpenSSL and Bouncy Castle. Both libraries include algorithms that are certified under the following consolidated CAVP certificates:

a) OpenSSL FIPS library version 3.0.13b-42 under CAVP Certificate #A7369

b) BC-FJA (Bouncy Castle FIPS Java API) Software Version 2.0 under CAVP Certificate #A7362

The following tables contain the CAVP algorithm certificates for the two cryptographic libraries implemented in the TOE:

SFR(s) Supported	Algorithm(s) (cryptographic operation)	Standard	CAVP Algorithm List Name	CAVP Cert. #
FCS_CKM.1	Key Generation RSA (2048 bits or greater)	NIST FIPS 186-4	RSA KeyGen	A7369
	ECC schemes using 'NIST curves' P-256, P-384, P-521	NIST FIPS 186-4	ECDSA KeyGen KeyVer	A7369
FCS_CKM.2	Key Establishment RSAES-PKCS v1_5	RFC 8107 Section 7.2	Vendor Affirmation	N/A
	Elliptic curve-based Key Establishment	NIST Special Publication 800-56A Revision 3	KAS-ECC-SSC Sp800-56Ar3	A7369
FCS_COP.1/DataEncryption	AES Encryption/Decryption AES-CTR (128, 256 bits) AES-GCM (128, 256 bits)	NIST SP 800-38A	AES-CTR AES-GCM	A7369
FCS_COP.1/SigGen	Sig Generation/Verification RSA 2048 bits	NIST FIPS 186-4	RSA SigGen SigVer	A7369
FCS_COP.1/Hash	Cryptographic Hashing SHA-256, SHA-384 SHA-512 Digest sizes 160, 256, 384, 512	NIST FIPS 180-4	SHA	A7369
FCS_COP.1/KeyedHash	Keyed Hash Algorithm HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 Key Sizes 160, 256, 384, 512 bits Digest Sizes 160, 256, 384, 512	NIST FIPS 198-1	HMAC	A7369
FCS_RBG_EXT.1	Random Bit Generation CTR_DRBG (AES-256) with software-based noise sources with minimum of 256 bits of entropy	NIST SP 800-90A	Counter DRBG	A7369

Table 15: Cryptographic Algorithm Table for OpenSSL

SFR(s) Supported	Algorithm(s) (cryptographic operation)	Standard	CAVP Algorithm List Name	CAVP Cert. #
FCS_CKM.1	N/A	N/A	N/A	N/A
FCS_CKM.2	Key Establishment RSAES-PKCS v1_5	RFC 8107 Section 7.2	Vendor Affirmation	N/A
FCS_COP.1/DataEncryption	AES Encryption/Decryption AES-CBC (128, 256 bits) AES-GCM (256 bits)	NIST SP 800-38A	AES-CBC AES-GCM	#A7362
FCS_COP.1/SigGen	Sig Generation/Verification RSA 2048 bits	NIST FIPS 186-4	RSA SigGen SigVer	#A7362
FCS_COP.1/Hash	Cryptographic Hashing SHA-1, SHA-256, SHA-384, SHA-512 Digest sizes 160, 256, 384, 512	NIST FIPS 180-4	SHA	#A7362
FCS_COP.1/KeyedHash	Keyed Hash Algorithm HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 Key Sizes 160, 256, 384 bits Digest Sizes 160, 256, 384	NIST FIPS 198-1	HMAC	#A7362
FCS_RBG_EXT.1	Random Bit Generation Hash DRBG (256, 384, 512) with software-based noise sources with minimum of 256 bits of entropy	NIST SP 800-90A	Hash DRBG	#A7362

Table 16: Cryptographic Algorithm Table for Bouncy Castle

8.2.1 FCS_CKM.1

OpenSSL provides the asymmetric key generation services for TLS and SSH. Bouncy Castle is not used for asymmetric key generation services.

The TOE implements a FIPS PUB 186-4 conformant RSA key generation mechanism for establishing TLS connections. Specifically, the TOE's implementation of RSA key generation complies with FIPS 186-4 (Digital Signature Standard (DSS) Appendix B.3) supporting a 2048-bit key size.

The TOE implements a FIPS PUB 186-4 conformant ECC schemes key generation mechanism for establishing SSH connections. Specifically, the TOE's implementation of ECC key generation complies with FIPS PUB 186-4, Digital Signature Standard (DSS), Appendix B.4 using NIST curves P-256, P-384, and P-521. See Tables 16 & 17 Cryptographic Algorithm Tables for OpenSSL and Bouncy Castle certification numbers.

NOTE: TOE is not claiming support for FIPS PUB 186-5.

8.2.2 FCS_CKM.2

The TOE implements RSA key establishment, conformant to RSAES-PKCS1-v1_5 in support of the TOE's TLS client and TLS server services (FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1). The TOE complies with section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography

Specifications Version 2.1 and all subsections regarding RSA key pair generation and key establishment in RSAES-PKCS1-v1_5. The TOE uses OpenSSL to generate RSA key pairs with a modulus of at least 2048 bits which has an equivalent key strength of 112 bits.

The RSA key establishment is used for TLS communications for remote administration using the Console, exporting audit data to the audit server, and authentication requests to external authentication server.

In addition, the TOE implements an Elliptic curve-based key establishment scheme that meets NIST Special Publication 800-56A Revision 3 in support of the TOE SSH server services (FCS_SSHS_EXT.1). The Elliptic curve-based support is provided by OpenSSL.

Below is a table that summarizes which cryptographic library is supporting which claimed interface.

OE Component	Definition of Communication (protocol, client/server, cryptographic service)
Management Workstation	Communications are secured using TLS where the TOE is the Server. TOE crypto required to support interface E3 as defined in Figure 1 above. RSA Key Generation Cryptographic services for certificate creation: OpenSSL RSA Key Establishment and encryption services for TLS: Bouncy Castle
	Communications are secured using SSH where the TOE is the Server TOE crypto required to support interface E2 as defined in Figure 1 above. Key Generation Cryptographic services: OpenSSL Elliptic curve-based Key establishment and encryption services for SSH: OpenSSL
Active Directory Server	Communications are secured using TLS where the TOE is the client. TOE crypto required to support interface E6 as defined in Figure 1 above. RSA Key establishment and encryption services for TLS: OpenSSL
Audit Server	Communications are secured using TLS where the TOE is the client. TOE crypto required to support interface E7 as defined in Figure 1 above. RSA Key establishment and encryption services for TLS: OpenSSL

Table 17: Identification of Cryptographic Services Supporting Secured Communication Channel

See Tables 16 & 17 Cryptographic Algorithm Tables for OpenSSL and Bouncy Castle certification numbers.

8.2.3 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.

Name	Origin	Store	Zeroization / Destruction
Diffie-Hellman Shared Secret	SSH Server / client applications	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00)*. After overwriting, the TSF reads the memory to verify the key has been destroyed. If the read-verify fails, the process is repeated. The key is zeroized immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Automatically zeroized after DH exchange.
Diffie-Hellman private exponent	SSH Server / client applications	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00)*. After overwriting, the TSF reads the memory to

Name	Origin	Store	Zeroization / Destruction
			verify the key has been destroyed. If the read-verify fails, the process is repeated. The key is zeroized immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Automatically zeroized after DH exchange
SSH session key	SSH Server / client applications	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00)*. After overwriting, the TSF reads the memory to verify the key has been destroyed. If the read-verify fails, the process is repeated. The key is zeroized immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Automatic zeroized after SSH session is terminated.
SSH Server Host Private Key	Generated on platform during initial setup of device.	Filesystem	Filesystem: Generation of a new key will only be accomplished during a reinstallation of the product where all files would be overwritten which would in effect also destroy the abstraction that represented the key.
TLS Server Host Certificate Private Key	Generated on platform (OpenSSL) during initial setup or imported after installation. OpenSSL TLS Communications for audit server and AD Bouncy Castle TLS Communication for Console	RAM and Filesystem	RAM: The Server Certificate's private key is destroyed by a single direct overwrite consisting of zeroes (0x00)*. After overwriting, the TSF reads the memory to verify the key has been destroyed. If the read-verify fails, the process is repeated. The key is zeroized immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Filesystem: Private key is deleted when generation of a new certificate are imported or when certificates are removed. The TOE will invoke the interface File.delete(), provided by a part of the TSF, that instructs a the TSF to destroy the abstraction that represents the key (i.e. delete the resource).

Table 18: Crypto key destruction

*OPENSSL_cleanse() and Bouncy Castle: JVM garbage collection APIs that perform zeroization

8.2.4 FCS_COP.1/DataEncryption

The TOE performs encryption and decryption using the AES algorithm in CTR, CBC, and GCM modes with key sizes of 128 and 256 bits. The AES algorithm meets ISO 18033-3, CTR and CBC meet ISO 10116 and GCM meets ISO 19772. The TOE's AES implementation is validated under CAVP. See Tables 16 & 17 Cryptographic Algorithm Tables for certification numbers.

- OpenSSL supports:
  - TLS communication: AES-GCM-128, AES-GCM-256

  - SSH communication: AES-CTR-128, AES-CTR-256, AES-GCM-128, AES-GCM-256
  - CTR DRBG: AES-CTR-256
- Bouncy Castle supports:
  - TLS communication: AES-CBC-128, AES-CBC-256, AES-GCM-256.

8.2.5 FCS_COP.1/SigGen

The TOE performs digital signature services generation and verification in accordance with RSA Digital Signature Algorithm (rDSA) with key sizes (modulus) 2048 bits. The RSA schemes are in accordance with 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. The TOE's RSA implementation is validated under CAVP. See Tables 16 & 17 Cryptographic Algorithm Tables for OpenSSL and Bouncy Castle certification numbers.

This is applicable to both cryptographic libraries being implemented.

NOTE: TOE is not claiming support for the FIPS PUB 186-5.

8.2.6 FCS_COP.1/Hash

The TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512 as specified in ISO/IEC 10118-3:2004 (FIPS PUB 180-4). The TOE's SHS implementation is validated under CAVP. See Tables 16 & 17 Cryptographic Algorithm Tables for OpenSSL and Bouncy Castle certification numbers. This is applicable to both cryptographic libraries being implemented. The hashing function is used to support password hashing of all passwords stored on the TOE (FPT_APW_EXT.1: SHA-256), Trusted updates digital signature verification (FPT_TUD_EXT.1: SHA-256), and TSF self-testing hash value check verification (FPT_TST_EXT.1: SHA-256). The hashing function supports the keyed hashed requirements for TLS algorithm (OpenSSL: SHA-256, SHA-384 BC: SHA-1, SHA-256, SHA-384) support and for the HASH_DRBG(SHA-256, SHA-384, SHA-512) used by Bouncy Castle. Additionally, the hashing function supports the hash requirements for rsa-sha2-256 publickey for SSH communications (OpenSSL SHA-256).

- OpenSSL supports: SHA-256, SHA-384, and SHA-512
- Bouncy Castle supports: SHA-1, SHA-256, SHA-384, and SHA-512

8.2.7 FCS_COP.1/KeyedHash

The TOE provides keyed-hashing message authentication services that meet ISO/IEC 9797-2:2011 (FIPS PUB 198-1, and FIPS PUB 180-4), Section 7 "MAC Algorithm 2". The TOE supports the following:

- HMAC-SHA-1 [key-size: 160 bits, digest size: 160 bits, block size: 512 bits, MAC lengths: 160 bits] for TLS communication support (BC: Console)
- HMAC-SHA-256 [key-size: 256 bits, digest size: 256 bits, block size: 512 bits, MAC lengths: 256 bits] for SSH and TLS communication support (OpenSSL: AD, Syslog, SSH and BC: Console)
- HMAC-SHA-384 [key-size: 384 bits, digest size: 384 bits, block size: 1024 bits, MAC lengths: 384 bits] for TLS communication support only (OpenSSL: AD, Syslog, SSH and BC: Console)
- HMAC-SHA-512* [key-size: 512 bits, digest size: 512 bits, block size: 1024 bits, MAC lengths: 512 bits] for SSH communication support only (OpenSSL: SSH)

The TOE's HMAC implementation is validated under CAVP. See Tables 16 & 17 Cryptographic Algorithm Tables for OpenSSL and Bouncy Castle certification numbers.

Additionally, the TOE supports AES in GCM mode for SSH, where the keyed-hashing message authentication is declared as "implicit".

*Only OpenSSL provides HMAC-SHA-512 keyed-hashing message authentication. Meaning:

- OpenSSL supports: HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512
- Bouncy Castle supports: HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-384.

8.2.8 FCS_RBG_EXT.1

The TOE implementation of Bouncy Castle uses a hash deterministic random bit generator (Hash_DRBG). The TOE implementation of OpenSSL uses a counter mode random bit generator (CTR DRBG). Both DRBG used by the TOE are 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 one software-based noise-based 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. For example, a 256-bit AES key generation operation will collect at least 256 bits of entropy before the DRBG is invoked.

Both Bouncy Castle and OpenSSL collect entropy from CPU Jitter RNG version 3.4.1. The entropy source is described in greater detail in the proprietary Entropy Assessment Report which shows that 256-bits of output from the entropy source has 256-bits of full entropy.

The TOE's DRBG implementation is validated under CAVP. See Tables 16 & 17 Cryptographic Algorithm Tables for OpenSSL and Bouncy Castle certification numbers.

8.2.9 FCS_SSH_EXT.1 and FCS_SSHS_EXT.1

The TOE acts as an SSHv2 server for remote CLI sessions that complies with RFCs 4251, 4252, 4253, 4254, 4344, 5647, 5656, and 6668. The TOE implementation of SSH supports public key-based and password-based user authentication. SSH is used for remote Security administrators to connect securely to the TOE for CLI connections. If a public key is presented for user authentication, the TOE will verify that the SSH client's presented public key matches one that is stored within the SSH server's authorized keys database. If the SSH client's presented public key does not match a stored key on the TOE, the TOE will consider this a failed authentication attempt and the connection will not be established. In the case of a password-based authentication attempt, the presented user credentials are verified using the TOE's native authentication mechanism. If the presented user credentials cannot be verified, then the connection will not be established. When establishing the SSH connection, the TOE utilizes a SSH Key Derivation Function in accordance with RFC 5656 to derive session keys from the shared secret.

The SSH implementation will detect all large packets greater than 262,130 bytes and drop accordingly. Additionally, the TSF enforces the connection to be rekeyed after no longer than one hour, no more than one gigabyte of transmitted data, or no more than one gigabyte of data is received, whichever threshold is reached first. The SSH rekey time and size threshold parameters are not administratively configurable.

The TOE's implementation of SSHv2 only supports:

- aes128-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com for its encryption algorithms
- rsa-sha2-256 as the only public key algorithm (user and host authentication)
- hmac-sha2-256, hmac-sha2-512, and implicit for data integrity
- ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 for key exchange method

When the TOE uses AES in GCM mode for SSH, the keyed-hashing message authentication is "implicit" through the selection of AES-GCM for the data integrity MAC algorithm.

OpenSSL provides all cryptographic support required for SSH communication.

8.2.10 FCS_TLSC_EXT.1

The TOE, when acting as a TLS client, will only support the TLSv1.2 protocol and rejects all other versions of TLS to connect and secure the following trusted channels:

- performing authentication requests with the AD Server
- audit data transfer to the Audit Server

OpenSSL provides the cryptographic support for key establishment and encryption for these TLS channels when TOE acts as client.

In the evaluated configuration, the ciphersuites utilized by the TOE cannot be configured. The following ciphersuites are used for the connections to both the AD and the audit server:

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

The TOE supports only NIST curves secp256r1, secp384r1, and secp521r1. These curves are configured by default and are the same for the AD and audit server connections. The TSF presents the Supported Groups Extension with the secp256r1, secp384r1, and secp521r1 curves/groups in the Client Hello.

The signature algorithms presented by the TOE are configured by default and are the same for the AD and audit server connections. The TOE presents the signature_algorithms extension with support for the following algorithms:

- rsa_pkcs1 with sha256(0x0401)
- rsa_pkcs1with sha384(0x0501)
- rsa_pkcs1 with sha512(0x0601)

The TOE will only establish a trusted channel if the peer certificate is valid. The TSF shall verify the presented identifier matches the reference identifier according to RFC 6125. The Common Name and Subject Alternative Name (DNS Name only) are the only reference identifiers in the certificate that are part of that validation.

The following functionalities are not supported by the TOE:

- Mutual authentication
- TLS renegotiation attempts
- PSKs
- Early data extension

- Post-handshake client authentication

The TOE will only support a wildcard in the left-most label (e.g. *.example.com). All other usages of a wildcard will cause a failure in the connection. The TOE does not support URI, IP addresses, service name reference identifiers, or pinned certificates.

8.2.11 FCS_TLSS_EXT.1

The TOE, when acting as a TLS server, will only support TLSv1.2 protocols to connect and secure the following trusted channels:

- Console remotely connecting to the TOE for remote management

Bouncy Castle provides the cryptographic support for key establishment and encryption TLS channel when TOE acts as server.

In the evaluated configuration, the ciphersuites utilized by the TOE cannot be configured. The following ciphersuites are used for the evaluated configuration:

- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288

The TOE will deny connections from a client requesting SSL 2.0, SSL 3.0, TLSv1.0, TLSv1.1, and TLSv1.3 protocol versions. When the TOE receives a TLS connection request with the wrong (unsupported) version, it returns a Fatal Alert: Handshake failure message and terminates the connection. The TSF supports authentication using X.509 certificates utilizing only RSA with 2048 bit key size and key exchange using RSA key establishment with 2048 bit key size. Session resumption based on session IDs according to RFC 5246 (TLS 1.2) is supported. The TOE only supports single context resumption based on session IDs, meaning that only the specific TOE from the initial connection with the client can determine if resumption will be allowed. The client and TOE must complete a full initial TLS handshake and generate a unique session ID. The TOE stores the session's negotiated parameters (for example the master secret and cipher suite) in its local memory or cache, associating it with the session ID. Additionally, the client stores the session ID. When the client reconnects, it sends the stored session ID in its "ClientHello" message where the TOE then looks for a matching session ID in its local cache. If a match is found, the TOE and client will re-establish the secure connection without a full handshake. If a match can't be found, the TOE rejects the attempts and replies with an Alert: Fatal Error. A client would have to complete a new initial connection with the TOE.

The following functionalities are not supported by the TOE:

- Session tickets
- Mutual authentication
- Early data extension
- PSKs
- TLS renegotiation attempts.

8.3 Identification and Authentication

8.3.1 FIA_AFL.1

The TSF implements separate configurable counters for tracking consecutive failed remote authentication attempts per user account. CLI user accounts are separate from Console user accounts, meaning a CLI user cannot log into the Console and vice versa. The TSF will lock the offending user account when the failure counter threshold is reached. A valid login that happens prior to the failure counter reaching its threshold will reset the counter to zero. To prevent total Security Administrator lockout, the TOE has a default cliadmin_console user that is only accessible through the TOE's physical local interface and is not subject to the authentication lockout feature. This account provides the ability to unlock other CLI or Console user accounts that have been locked due to reaching the failed number of authentication attempts threshold.

If the failure counter threshold is reached, the offending account is locked, preventing the user from logging into the TOE through the interface associated with that account (Console or Remote CLI). The lockout counter will reset to zero after either the Security Administrator manually unlocks the account or the administratively defined lockout time period has elapsed.

Security Management of configurable parameters:

Threshold setting

The Console Security Administrator configures the number of failed attempts lockout threshold through the Console. The threshold can be set to a minimum of 1 and maximum of 10 consecutive failed attempts and applies to both the CLI and Console users. The default setting is 3 consecutive failed attempts. A single counter is used for tracking a Remote CLI user's failed password-based and public-key based authentication attempts.

Separate automatic unlock time unlock settings

The Console Security Administrator is able to configure a time period when locked Console accounts will automatically unlock. The default for this setting is 30 minutes for the Console. The lockout time period can be configured between 5-1000 minutes.

The CLI Security Administrator is able to configure the time period for when a locked CLI account will automatically unlock. The default for this setting is 24 hours for the CLI users. The lockout time period can be configured between 1-1000 minutes using the "fstool set_property os.lockout.fail <time in seconds>" command.

Manual unlocking of accounts

For Console user accounts that are locked: A locked Console user account can be manually unlocked by the Console Security Administrator by navigating to the "Tools" > "Options" > "CounterACT User Profiles" page in the Console, selecting the locked user account, and pressing the activated "Unlock" button. Additionally, a CLI Security Administrator may unlock a Console user account by using the "fstool unlock_console_user <user-id>" command.

For CLI user accounts that are locked: A locked Remote CLI user account can be manually unlocked by the CLI Security Administrator by using the "fstool user faillock reset <locked username>" command.

8.3.2 FIA_PMG_EXT.1

The TOE supports the ability for a Console Security Administrator to set the minimum password length to 6 characters or greater with a maximum of 100 characters. Passwords can be composed of any combination of upper and lower-case letters, numbers and special characters. The accepted special characters include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")".

8.3.3 FIA_UAU.7

When authenticating to the TOE with a local physical connection (local CLI) to access the CLI, the password is obscured by suppressing the echo of keystrokes to the screen. No indication of progress is provided while typing in a password. Also, in the case of an invalid username or password, the TOE does not reveal any information about the invalid component.

8.3.4 FIA_UIA_EXT.1

The warning banner text can be configured by the Security administrator. The display and acknowledgement of this banner is the only TOE functionality that is available to an unauthenticated user.

When connecting to the TOE remotely using an SSH client (remote CLI) or using a local physical connection (local CLI) to gain access to the CLI, the TOE displays the pre-authentication warning banner. Users are authenticated using a native username/password credential authentication mechanism for local physical connections and SSH connections. SSH connections also support public key-based user authentication where the presented public key is compared to the Administratively imported public key assigned per user. Access is only granted once the user provides a valid username/password or a provides a correct public key.

When connecting to the TOE remotely using the Console, which establishes a TLS connection, the TOE displays the pre-authentication warning banner. The TOE can be configured to request an authentication decision from an Active Directory server or use the native username/password credential authentication mechanism for users connecting to the TOE using the Console.

Access is only granted once the user provides a valid username/password that is verified using Active Directory or native username/password credential authentication mechanism.

8.3.5 FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, and FIA_X509_EXT.3

The TOE uses X.509v3 certificates to support authentication for TLS connections to external IT entities in accordance with RFC 5280. The TOE performs certificate validity checking for all outbound TLS connections immediately upon receipt from the server during the TLS handshake. The TSF supports validation of certificate path lengths of three or more certificates.

When the TSF cannot determine the validity of a certificate, the TSF will not accept the certificate and not establish a connection. The TSF does not provide a mechanism to override the validation decision.

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 TSF treats a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE

- The certificate path must terminate with a trusted CA certificate. This CA certificate must be installed on the TOE and designated as a trust anchor.
- 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 certificate revocation status, when the certificate is used for the authentication step of the TLS handshake, using the Online Certificate Status Protocol (OCSP) as specified in RFC 6960. This includes the leaf certificate and all intermediate certificates received.
- When the TSF cannot establish a connection to determine the validity of a certificate the TSF does not accept the certificate and denies the connection.
- The TSF validates the extendedKeyUsage field according to the following rules:
    - Server certificates presented for TLS must have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses must have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
    - Client Authentication and Code Signing purpose key usages are not supported as the TOE does not use X509 certificates to sign updates and the TOE is not claiming support for mutual authentication.

The TSF will only perform a revocation status check after the successful validation of the certificate. If the presented certificate chain is determined to be invalid, the TSF terminates the attempted connection immediately and a revocation check is not performed

The TOE provides the functionality for a Security Administrator to generate a Certificate Request as specified in RFC 2986. The generation of the Certificate Request requires “Common Name, Organization, Organizational Unit, and Country” fields to be provided. The chain of certificates is validated from the root CA when the CA Certificate Response is received.

In order for the TOE to authenticate to the remote audit server and Active Directory servers, trusted CA certificates must be imported into the TOE’s certificate trust store via the Console.

8.4 Security Management

8.4.1 FMT_MOF.1/ManualUpdate, FMT_MTD.1/CoreData, FMT_MTD.1/CryptoKeys, and FMT_SMF.1

The SFRs listed above have been combined to clarify the Security Management functions of the TOE including how the TOE implements authentication, identification, and also RBAC. The following description will also include restrictions for these roles and functions.

The TOE uses role-based access control (RBAC), as described in FMT_SMR.2, to restrict access to the functions that manage the TSF data. The available functionality that is presented to an authenticated user is based on the group of permissions and the privileges associated with the permissions. These permissions/privileges are bound to the user only after the user has successfully authenticated. Display and acknowledgement of a warning banner is the only TOE functionality available prior to identification and

authentication. The Console limits the presented functionality based on the privileges bound to that user. The TSF restricts the ability to manage the TSF data to only Security Administrators.

The TSF management functions that are restricted to Security Administrators based on local or remote administration, and scoped by this evaluation are:

Management Function	Local CLI (physical connection)	Remote CLI (SSH)	Console (TLS)
Ability to administer the TOE remotely		X	X
Ability to configure the access banner			X
Ability to configure the remote session inactivity time before session termination			X
Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates			X
Ability to re-enable an Administrator account	X	X	X
Ability to set the time which is used for time-stamps	X	X	
Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors (i.e., generate, import, and delete X.509 certificates)			X
Ability to generate Certificate Signing Request (CSR) and process CA certificate response			X
Ability to administer the TOE locally	X		
Ability to configure the local session inactivity time before session termination or locking			X
Ability to configure the authentication failure parameters for FIA_AFL.1	X	X	X
Ability to manage the trusted public keys database (i.e., generate, import, and delete SSH keys)	X	X	

Table 19: Management Functions to Management Interface Identification

8.4.2 FMT_SMR.2

There are two types of user accounts, those that access the TOE through the CLI interfaces, and those that access through the Console. The TOE maintains the role of Security Administrator which is fulfilled by the following roles on each interface:

- Local CLI: "cliadmin" role and "cliadmin_console"
- Remote CLI: "cliadmin" role
- Console: "administrator" role by applying 'select all' permissions for the user account

The TOE is designed to use permissions which allow, limit or prevent user access to specific Console tools (access to the management functions available through the Console). Upon successful authentication, the TSF associates the administratively defined set of permissions (role) for that user to the subject acting on behalf of that user. The TSF then enforces role-based access control (RBAC) to limit access to TSF functions and data based on the set of permissions bound to the subject.

A Console user assigned the TOE's "administrator" role has access to all Console tools and features and is able to administer the TOE remotely as a Security Administrator. All other Console users that do not have the full set of administrative permissions are categorized as a "Console User" and are not Security Administrators of the TOE.

The TOE has one predefined Console administrative user called “Admin”. The “Admin” account is assigned the “administrator” role and these permissions cannot be modified or customized. A customized password must be created during installation by the customer. The “Admin” account is used to create additional Console Security Administrators.

The TOE has a dedicated account called ”cliadmin_console”. This account must be created during the configuration of the TOE. Its primary purpose is to recover locked-out users, including cliadmin, as it is not subject to authentication lockout. The cliadmin_console user can only log into TOE’s physical local interface and cannot remotely access the TOE. Its privileges are fixed, but the password can be changed.

For both the Local CLI and the Remote CLI interface, the TOE has an administrative user role “cliadmin”. A user assigned the “cliadmin” role can perform a limited number of administrative actions controlled by the on the TOE, meaning the Console interfaced and the CLI have different administrative functionality available.

8.5 Protection of the TSF

8.5.1 FPT_APW_EXT.1

No passwords are stored by the TOE in plaintext. All Console user passwords are hashed using SHA-256 and then encrypted using AES-256 before storing in an internal database. CLI user passwords are hashed using SHA-512. There is no function provided by the TOE to display a password value in plaintext.

8.5.2 FPT_SKP_EXT.1

The TOE does not provide a mechanism to view pre-shared keys, symmetric keys and private keys. Volatile memory used to store secret keys, private keys, and secret key data is not accessible by Security administrators and neither is the file system of the OS. Key data stored on the TOE are encrypted using AES-256. There are no keys stored in plaintext.

8.5.3 FPT_STM_EXT.1

The TOE appliance uses a hardware-based real-time clock based on a quartz crystal running at high frequency and accuracy. The TOE’s embedded OS manages the clock and provides the CLI Security Administrator the ability to set the time manually.

The following NDcPP defined security related functions obtain the time stamp from the TOE’s internal clock:

- Audit record timestamps (seconds, milliseconds, microseconds, or nanoseconds).
- X.509v3 certificate validation
- Inactivity of remote sessions
- Inactivity of local session

8.5.4 FPT_TST_EXT.1

Upon the startup of the TOE, multiple Power-On Self-Tests (POSTs) are run. The POSTs provide environmental monitoring of the TOE’s components (hardware and software), in which early warnings can prevent whole component failure.

The following self-tests are performed to verify the integrity of the software and cryptographic modules. The self-tests will also be run on service restarts and are available for manual execution. The following tests are part of the self-test suite:

#	Component	Validation	Fail Result
1.	Kernel	HMAC + Built-in Crypto Self-test	Hard-fail
2.	Core OS and packages (including OpenSSH)	Built-in RPM Verification	Hard-fail
3.	fipscheck utility	HMAC verified against fipshmac	Hard-fail
4.	Crypto: OpenSSL	fipscheck (including OpenSSL self-check)	Hard-fail
5.	OpenSSL rpm package	Built-in RPM Verification	Hard-fail
6.	Crypto: Bouncy Castle	Built-in crypto package self-test (KAT)	Hard-fail
7.	Core Platform and plugin installation packages and extracted files.	SHA-256 verified against last known or stored hash.	Soft-fail
8.	System current state vs system configuration	Running kernel version compared to version defined in grub; FIPS mode running status compared to configuration in grub.	Soft-fail

Table 20: Self-Test List with Failure Results

- **Hard-fail**: Kernel test failure will result in panic the OS. The machine will shutdown.
- **Soft-fail**: Upon test failure, the function would alert the local CLI Security Administrator upon login, write an audit event and send the audit record to the external audit server (if configured). The TOE will enter a maintenance mode by preventing the main TOE service from starting (i.e. not available for operational use), and an alert will be displayed on the local CLI.

Self-test number 1, 2, 3, 5, 7, and 8 occur during initial startup and self-test number 4 and 6 occur either on demand or prior to providing cryptographic services. A CLI Security Administrator may execute the self-test check manually using the `selftest` command. The output will be displayed to the screen in the following format:

selftest:144141:1628543527.855930:Mon Aug 9 17:12:07 EDT -0400 2021: Started
selftest:144141:1628543527.856168:Mon Aug 9 17:12:07 2021: Verifying fipscheck
selftest:144141:1628543527.912380:Mon Aug 9 17:12:07 2021: Verifying grub
selftest:144141:1628543527.936682:Mon Aug 9 17:12:07 2021: Verifying rpm: kernel (64-bit)
etc.

An example of an error discovered on a plugin check:

selftest:144141:1628543550.799895:Mon Aug 9 17:27:20 2021: problem plugin: hwi, plugin/hwi/scripts/hwi_cert_store_new.exe, file sha256sum, 6e23399aabb23038e07151c67b2c9008753509ee2d675b4a0d81d63744590c04 != c6f0d923ce293167507206795b3f3b982e6c8057a1929231f9ff86eb4753e9bf

These tests are sufficient to validate the correct operation of the TSF because they verify that both the base OS and TOE software have not been tampered with and that the underlying hardware does not have any anomalies that would cause the software to be executed in an unpredictable or inconsistent manner.

8.5.5 FPT_TUD_EXT.1

The Console Security Administrator can query the TOE for the currently executing version of the TOE software by going to the top menu bar, click the "Help" drop down menu, and then click "About Forescout".

The TOE does not automatically check for or download an update itself. When an update is available, a Security Administrator must download the update package to the management workstation. Once the update is on the management workstation the Console Security Administrator must manually initiate the installation via the Console.

For an Appliance license: Upon execution of the upgrade command the Console Security administrator has the choice of the following option:

- Upload and Upgrade - Upload the file to the device and begin the upgrade.

For an Enterprise Manager license: Upon execution of the upgrade command the Console Security administrator has the choice of the following three options:

- Upload Only – Upload the file to the device but do not begin the upgrade
- Upload and Upgrade – Upload the file to the device and begin the upgrade
- Upgrade – Upgrade the device from the previously uploaded file.

The Console uploads the update package over the existing TLS path that is already established between the Console and the TOE. Only one upgrade package can be uploaded to the TOE device at a time. A second attempt to upload an upgrade package will result in the administrator being warned that this will overwriting the existing upgrade package. The following provide more details to each of the installation options identified above:

Upload Only – The TSF automatically verifies the update's digital signature during the upload process. The TSF uses a locally stored public key (on the machine) to verify update package authenticity. This key is installed as part of the initial software installation and cannot be modified or changed by an administrator. The TSF will delete the uploaded file if the digital signature is determined to be invalid for any reason. There is no means for an administrative override to continue the upload. Once the upload is complete and the digital signature is valid, the Console indicates its success.

Upgrade Only – When there is an upgraded package available, a Console Security Administrator can select the Upgrade Only option to initiate the installation. The TOE will re-verify the digital signature prior to initiating the installation. The TSF will not continue with the installation if the digital signature is determined to be invalid for any reason. There is no means for an administrative override to continue the installation. Once the device has been upgraded, the device will reboot automatically, the upload storage area is emptied (meaning another upgrade package can now be uploaded but not installed), and the current operating version will be updated to reflect the recent upgrade version.

Upload and Upgrade – The TSF automatically verifies the update's digital signature during the upload process. Once the upload is complete and the digital signature is valid, the installation will begin. The TOE will re-verify the digital signature prior to initiating the installation. The TSF will not continue with the installation if the digital signature is determined to be invalid for any reason.

There is no means for an administrative override to continue the installation. Once the device has been upgraded, the device will reboot automatically, the upload storage area is emptied (meaning another upgrade package can now be uploaded but not installed), and the current operating version will be updated to reflect the recent upgrade version.

8.6 TOE Access

8.6.1 FTA_SSL_EXT.1

When a local session is inactive for the configured period of time, the TOE will terminate the session. The inactivity timer is configured by the Console Security Administrator via the Console and is set in minutes or hours.

8.6.2 FTA_SSL.3

The TOE will terminate a remote session due to inactivity according to the configuration threshold set by the Console Security Administrator. The inactivity timer is configured by the Console Security Administrator via the Console and is set in minutes or hours.

8.6.3 FTA_SSL.4

All users accessing the TOE is capable of terminating their own session. A Console user terminates their own current session by clicking the “Log-Out” command from the File menu. A CLI user terminates their own current session by typing "exit" at the command line.

8.6.4 FTA_TAB.1

There are three possible administrative ways to log into the TOE: locally via physical connection to access the CLI, remotely via SSH connection to access the CLI, and remotely using the Console which establishes a TLS connection. When logging in locally or remotely, the pre-authentication banner is displayed and is viewed prior to authentication. The authentication banner is administratively customizable by the Console Security Administrator via the Console.

8.7 Trusted Path/Channels

8.7.1 FTP_ITC.1

The TOE provides the ability to secure sensitive data in transit to and from the Operational Environment. The TOE, acting as the TLS client, uses the TLS protocol to initiate and establish the trusted channel to support the following capabilities:

- to export audit data to an audit server
- authenticate users via an Active Directory server

The TOE’s TLS client implementation is conformant to FCS_TLSC_EXT.1. TLS communications use X.509v3 certificates to support authentication.

8.7.2 FTP_TRP.1/Admin

Remote administration is secured by using SSH and TLS protocols.

The Console establishes the TLS connection to the TOE on behalf of the user for remote administration. The TOE is acting as a TLS server and is conformant to FCS_TLSS_EXT.1. The Console is using the host platforms TLS client capabilities.

A user can connect to the TOE using SSH to remotely manage the TOE via the CLI (remote CLI). The TOE's SSH server implementation is conformant to FCS_SSHS_EXT.1.