Forescout eyeInspect v5.2

Security Target

ST Version: 1.0
November 29, 2024

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.

1.1.1 ST Identification

ST Title: Forescout eyeInspect v5.2 Security Target
ST Version: 1.0
ST Publication Date: November 29, 2024
ST Author: Booz Allen Hamilton

1.1.2 Document Organization

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

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

Chapter 3 describes the conformance claims made by this ST.

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

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

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

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

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

1.1.3 Terminology

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

Term	Definition
Target of Evaluation (TOE)	A set of software, firmware and/or hardware possibly accompanied by guidance. For this document, the TOE is the evaluated Forescout eyeInspect product configured to meet its security claims.
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.).
TOE Security Function (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the SFRs.
TSF Data	Data for the operation of the TSF upon which the enforcement of the requirements relies.

Table 1: CC Specific Terminology

Term	Definition
Security Administrator	The class of TOE administrators that are tasked with managing the TOE's functional and security configuration. Embodies those administrators that have access to the local and remote administrative interfaces. For the local and remote CLI, the Security Administrator is the *silentdefense* user. For the Web GUI, the Security Administrator is the Admin user.
Command Center	A component of Forescout eyeInspect used for collecting and processing data reported by the Sensors in a deployment of Forescout eyeInspect. The Command Center also supports a web interface for management of the TOE.
Local Command Line Interface (CLI)	The local CLI is utilized to perform administrative management functions on the Command Center or Sensor at the base operating system level. This interface is accessible through a terminal that is connected directly to the product's Command Center or Sensor component.
Remote CLI	The remote CLI is utilized to perform administrative management functions on the Command Center or Sensor at the base operating system level. This interface is accessible over a secure SSH trusted channel from a management workstation to the product's Command Center or Sensor component.
Remote Management Workstation	A standard PC used for remote access to the TOE via either the Web GUI (HTTPS) or remote CLI (SSH).
Sensor	A component of Forescout eyeInspect used for monitoring Industrial Control System and Supervisory Control and Data Acquisition (ICS/SCADA) networks that it is deployed in. The Sensor is a managed component of the Command Center.
Terminal	The device that is connected directly to the appliance through the keyboard/video ports or a serial port. The device will act as a terminal emulator that is compatible with serial communications used for access to the local CLI.
Web Graphical User Interface (GUI)	The Web GUI is utilized to perform administrative management functions on the Command Center. This interface is accessible over a secure HTTPS trusted channel from a management workstation to the product's Command Center.

Table 2: Customer Specific Terminology

1.1.4 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
CA	Certificate Authority
CC	Common Criteria
CLI	Command-line Interface
CPU	Central Processing Unit

Acronym	Definition
DB	Database
DRBG	Deterministic Random Bit Generator
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
ICS	Industrial Control System
IT	Information Technology
NDcPP	Collaborative Protection Profile for Network Devices
NIAP	National Information Assurance Partnership
NTP	Network Time Protocol
OE	Operation Environment
OS	Operating System
OT	Operational Technology
PP	Protection Profile
RAM	Random Access Memory
RBAC	Role-Based Access Control
RU	Rack Unit
SAR	Security Assurance Requirement
SCADA	Supervisory Control and Data Acquisition
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

Table 3: Acronym Definition

1.1.5 Reference

[1] collaborative Protection Profile for Network Devices Version 2.2e 20200323 [NDcPP]

[2] 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

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

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

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

1.2 TOE Reference

The TOE is Forescout eyeInspect v5.2 which consists of a Command Center and one or more Sensors. The TOE contains the following models for each TOE component:

- Command Center: Forescout FS-HS-5160-OT
- Sensors: Forescout FS-HW-5120, Forescout FS-HW-5160, Forescout FS-HW-4130, and Forescout FS-HW-2130

The minimum configuration for a deployment of Forescout eyeInspect is one Command Center and one Sensor. Only one Command Center can be deployed as part of the operational configuration. Including additional Sensors within a deployment of Forescout eyeInspect as part of the operational configuration will not affect the validity of the functional claims made within this document and the Common Criteria certification.

1.3 TOE Overview

Forescout eyeInspect's primary purpose is to help reduce risk, automate compliance, and optimize threat analysis for industrial operations management technology within a network. The Command Center provides the main interface for management of eyeInspect, including the ability to manage eyeInspect configuration, manage Sensors, and perform analytics on collected device and threat data. Meanwhile, eyeInspect Sensor(s) receive device information gathered from within the network and send it to the Command Center for analysis. A Forescout eyeInspect deployment consists of one Command Center and at least one Sensor. The Command Center and the Sensor(s) work together to provide visibility and an understanding of security posture for Industrial Control System and Supervisory Control and Data Acquisition (ICS/SCADA) networks. The collecting and analyzing functionality described above is not included within the scope of the evaluation. Only the functionality claimed in Section 6 of this Security Target is considered to be within the logical boundary of the TOE.

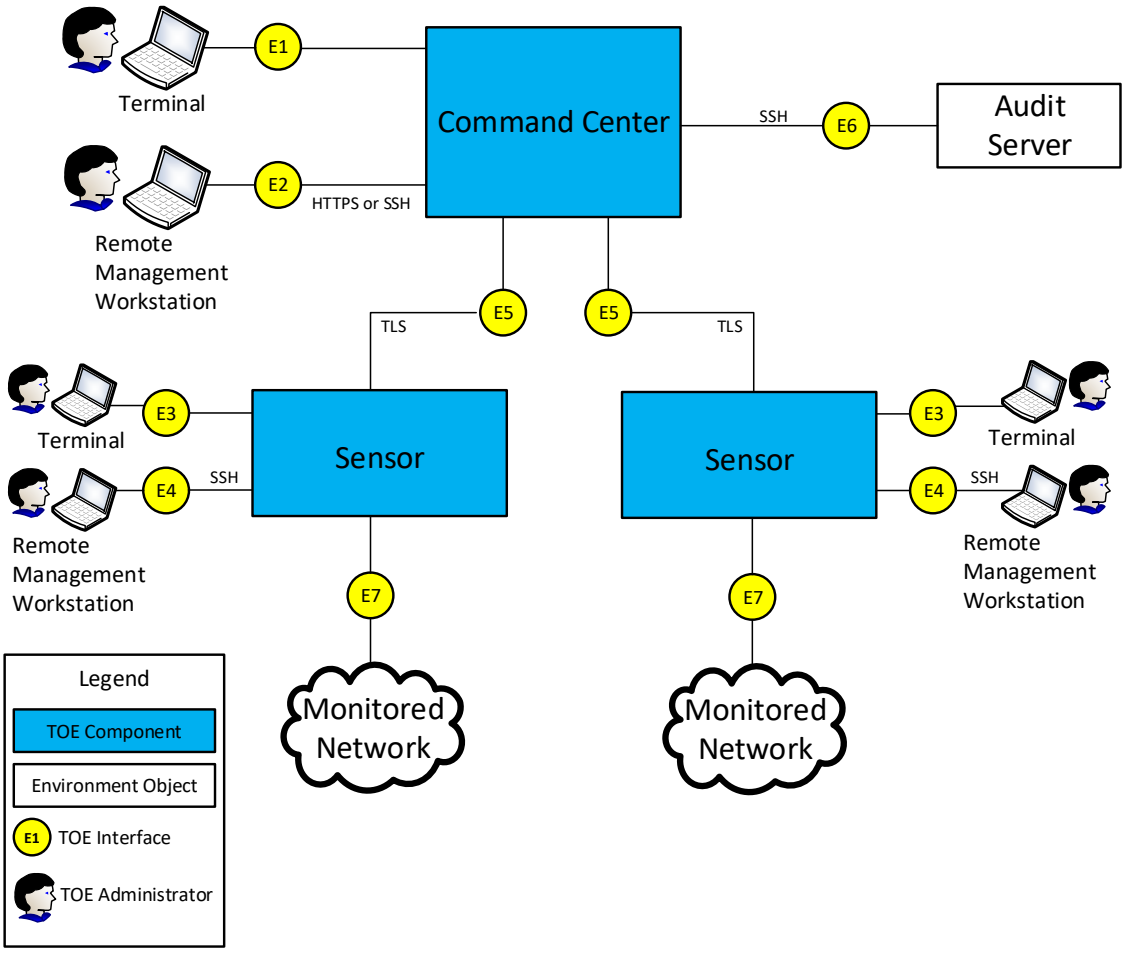

Figure 1: TOE Boundary for Forescout eyeInspect

The TOE boundary diagram depicted in Figure 1 above consists of one Command Center and two Sensors. The illustration is meant to serve as an example of one kind of deployment configuration. A deployment configuration with a single Sensor is also acceptable as well as adding more Sensors into the deployment. The Forescout eyeInspect Command Center and Sensors are separately installed appliances which together provide an administrator the ability to manage the monitored network. The connections to the TOE are as follows:

- **E1: Terminal to Command Center** – The Terminal utilizes a direct local connection to the Command Center through a designated management port. The Terminal provides a Command Line Interface (CLI) for local management of Command Center.
- **E2: Remote Management Workstation to Command Center** – The Remote Management Workstation allows a Security Administrator to access either the Web GUI or remote CLI for remote administration of the Command Center. These connections are bundled together in Figure 1 as connection E2.
    - The Command Center can be accessed via the Web GUI using a HTTPS connection over a standard browser. In this case, the Command Center acts as an HTTPS server to the standard browser used on the Remote Management Workstation.

  - The Command Center can be accessed via a remote CLI using a SSH connection. The Command Center acts as an SSH server to the SSH Client used on the Remote Management Workstation.
- **E3: Terminal to Sensor** – The Terminal provides a CLI for local management of a Sensor. The Terminal utilizes a direct local connection to the Sensor through monitoring ports.
- **E4: Remote Management Workstation to Sensor** – The Remote Management Workstation provides a remote management interface for a Sensor deployed in an operational configuration of the TOE. The Sensor acts as an SSH server to the SSH Client used on the Remote Management Workstation. There is no Web GUI available for remote administration of a Sensor.
- **E5: Sensor to Command Center** – A Sensor communicates with the Command Center via two secure TLS channel. These connections are used for the Command Center to manage a Sensor installed in a deployment of the TOE, send audit data from the Sensors to the Command Center, and to send collected network data from the Sensors to the Command Center.
- **E6: Command Center to Audit Server** – The Command Center communicates with an external Audit server via a secure SSH channel for external audit record storage.
- **E7: Sensor to Monitored Network** – The network interfaces involved in these connections have no IPv4 identity; traffic on the sensor end is only passively analyzed by the Forescout eyeInspect product to monitor the ICS/SCADA network. The interface is not related to any functionality claimed in Section 6 of this Security Target but is being described for completeness of understanding eyeInspect's interaction with the Operational Environment for its primary purpose.

1.4 TOE Type

The TOE type for the Forescout eyeInspect product is a Network Device. Forescout eyeInspect is used to help reduce risk, automate compliance, and optimize threat analysis for industrial operations management technology.

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… Under this cPP, NDs may be physical or virtualized. A physical Network Device (pND) consists of network device functionality implemented inside a physical chassis with physical network connections. The network device functionality may be implemented in either hardware or software or both. For pNDs, the TOE encompasses the entire device—including both the network device functionality and the physical chassis. There is no distinction between TOE and TOE Platform."

The eyeInspect product is distributed meaning that it is composed of distinct components that must work in tandem for it to meet its intended purpose. A deployment of the eyeInspect product requires one Command Center and one or more Sensors. The product cannot operate as intended without both components. Each eyeInspect component is composed of both hardware and software. When the eyeInspect product is connected into an enterprise's network, it provides visibility into the enterprise's network infrastructure and aligns with the requirements of a network device. Therefore, the distributed eyeInspect product claims conformance to all NDcPP requirements as claimed in Section 6 of this Security Target.

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 Component	Component Description	Hardware Model(s)	Software Version
Forescout eyeInspect Command Center	Used by a Security Administrator for TOE management and analyzing collected network data	Forescout FS-HS-5160-OT	Forescout eyeInspect Command Center v5.2
Forescout eyeInspect Sensor	Used to collect network data	Forescout FS-HW-5120, Forescout FS-HW-5160, Forescout FS-HW-4130, Forescout FS-HW-2130	Forescout eyeInspect Sensor v5.2

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
Terminal	A terminal is a device that handles the input and display of data when connected to an appliance's serial port. A terminal client, such as *Hyper Terminal* (Windows) or *minicom* (Linux) can be used on a general purpose computer. The TOE's CLI can be accessed locally with a physical connection to the TOE using the designated management port and must use a terminal emulator that is compatible (E1 & E3) or use the keyboard and display ports. The terminal client (emulator) must support the following parameters: • Baud: 19200 • Parity: None • Data Bit: 8 • Stop Bits: 1 • Flow Control: None (*minicom* enables flow control by default-edit its configuration to disable this) • Emulation: ANSI (at least for *minicom*)
Remote Management Workstation	Any general-purpose computer that is used by an administrator to manage the TOE. For the TOE to be managed remotely the Remote Management Workstation is required to have: • SSHv2 client installed to access the TOE's CLI on both TOE components • Web browser installed to access the Web GUI on the Command Center TCP communications from the Remote Management Workstation to the TOE is secured using: • SSH for remote access to the CLI

Component	Definition
	• HTTPS for remote access to the Web GUI The TOE acts as a server for both protocols. This component is required to support interfaces E2 & E4 as defined in Figure 1 above.
Audit Server	The TOE connects to an audit server to send the audit records for remote storage via SSH connection where the TOE is the SSH 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 E6 as defined in Figure 1 above.
Monitored Network	The monitored network contains operational technology components, Industrial Control Systems, Supervisory Control and Data Acquisition systems, etc. Figure 1 identifies these as a single interface. The interface to the manage the Forescout eyeInspect product is a separate connection from that of the monitored network that the Forescout eyeInspect product is managing. The Forescout eyeInspect's management of the monitored network is out of scope for the NDcPP. Therefore, interface E7 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 functionality evaluated is scoped exclusively to the security functional requirements specified in the Security Target.

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

2.3.1 Not Installed

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

This product contains functionality that is part of the purchased product but is not part of the TSF relevant functionality that is being evaluated as the TOE based on the Protection Profile. Non-PP functionality provided by these devices needs to be assessed separately and no further conclusions should be drawn about their effectiveness.

2.4 Physical Boundary

The following tables outline the models and their key differentiators that are part of the evaluation.

TOE Component Name	Equipment		
	Software/Firmware	**Hardware Model**	**Component/Configuration**
Forescout eyeInspect: Command Center	Forescout eyeInspect v5.2 operating on Linux	Forescout FS-HS-5160-OT	1U Desktop, 19" rack server
			CPU Xeon Gold 6132 2x 14C/28T

	Ubuntu 20.04.6 LTS OS		1/10 GB Network card
			8 Copper, 2 Fiber, 2 unused SFP Ports

Table 6: Forescout eyeInspect Command Center

TOE Component Name	Equipment		
	Software/Firmware	**Hardware Model**	**Component/Configuration**
Forescout eyeInspect: Sensors	Forescout eyeInspect v5.2 operating on Linux Ubuntu 20.04.6 LTS OS	Forescout FS-HW-5120	1U rackmount
			CPU Intel Xeon Silver 4114 2x 10C/20T (Skylake)
			1GB out of band management port
			4x 10/100/1000 Mbps Ethernet
			4x 1G/10G dual rate SR 2x Fiber SFPs included in base configuration
		Forescout FS-HW-5160	1U rackmount
			CPU Xeon Gold 6132 2x 14C/28T (Skylake)
			1GB out of band management port
			4x 10/100/1000 Mbps Ethernet
			4x 1G/10G dual rate SR 2x Fiber SFPs included in base configuration
		Forescout FS-HW-4130	1U rackmount
			Gen 8 Intel Core i5-8500T 6C/6T CPU 64bit (Coffee Lake S)
			2 x 10/100/1000 Mbps Ethernet (i210-IT & i219-LM)
			4 x 10/100/1000 Mbps Ethernet (i210-IT)
		Forescout FS-HW-2130	Shelf/desktop (31 x 100 x 125 mm.)
			Intel Celeron J3455 1.50 GHz 4C/4T CPU 64bit (Apollo Lake: Microarchitecture: Goldmont)
			2-4 x Intel 10/100/1000 Mbps Ethernet (i210AT)

Table 7: Forescout eyeInspect Sensors

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
- Communication
- 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 for all components of the TOE. Both the Command Center and the Sensor store audit logs locally. The TOE supports forwarding

audit records to an external audit server at a predefined frequency. There is no direct connection between the Sensors and the remote audit server. Therefore, audit events from the Sensors are first forwarded to the Command Center, and then forwarded to the remote audit server by the Command Center. The Command Center also forwards its audit records directly to the external audit server. In the evaluated configuration, the audit data is securely transmitted to the audit server using a SSHv2 communication channel.

2.5.2 Cryptographic Support

The TOE provides cryptography in support of TLS (v1.2), HTTPS, and SSH trusted communications. The Command Center utilizes Bouncy Castle for TLS and HTTPS communications, and OpenSSL for SSH communications. The Sensor utilizes OpenSSL for TLS and SSH communication. The TOE destroys keys when no longer needed. The following table identifies the cryptographic services per cryptographic library.

SFR		Command Center		Sensors
		Bouncy Castle	**OpenSSL**	**OpenSSL**
FCS_CKM.1	ECC using NIST curves P-256, per FIPS PUB 186-4	#A6120	#A6128	#A6128
	FFC using safe-prime groups NIST Special Publication 800-56A Revision 3 and RFC 3526.	N/A – Bouncy Castle does not provide FFC services	#A6128	#A6128
FCS_CKM.2	Elliptic curve-based key establishment NIST Special Publication 800-56A Revision 3	#A6120	#A6128	#A6128
	FFC using safe-prime NIST Special Publication 800-56A Revision 3 and groups listed in RFC 3526.	N/A – Bouncy Castle does not provide FFC services	#A6128	#A6128
FCS_COP.1/ DataEncryption	AES GCM 256 bits	#A6120	N/A	#A6128
	AES CTR 256 bits	N/A	#A6128	#A6128
FCS_COP.1/ SigGen	RSA FIPS 186-4 Signature Services 2048 bits	#A6120	N/A	#A6128
	ECDSA FIPS 186-4 Signature Services 256 bits	N/A	#A6128	#A6128
FCS_COP.1/ Hash	SHA-256	#A6120	#A6128	#A6128
	SHA-384	#A6120	N/A	#A6128
	SHA-512	#A6120	#A6128	#A6128
FCS_COP.1/ KeyedHash	HMAC-SHA-256	N/A	#A6128	#A6128
	HMAC-SHA-384	#A6120	N/A	#A6128
	HMAC-SHA-512	N/A	#A6128	#A6128
FCS_RBG_EXT.1	Hash DRBG	#A6120	N/A	N/A
	CTR_DRBG	N/A	#A6128	#A6128

Table 8: Cryptographic Services

2.5.3 Communication

Initial TLS communications between TOE components does not occur until the Sensor is configured and enabled by the Security Administrator. Once enabled the Sensor application will send a request for enrollment, via TLS, to the configured Command Center, where the Security Administer must approve before full communications are established.

2.5.4 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. Additionally, a Security Administrator can define the minimum password length. The displaying of a pre-authentication warning banner is the only function available prior to user authenticating.

The TOE provides a native password authentication mechanism for Web GUI and CLI users. The inter-TOE TLS client functionality on the Sensor performs the validation, without revocation checking, of the presented X.509v3 certificates from the Command Center server.

2.5.5 Security Management

The TOE uses role-based access control to prevent unauthorized management of and access to TSF data. The TOE provides a Security Administrator role that can be assigned to a user which provides the ability to administer the TOE locally and remotely.

2.5.6 Protection of the TSF

The TOE ensures the security and integrity of all data that is stored locally and accessed locally or remotely. User authentication passwords are not stored in plaintext. The Security Administrator is required to manually initiate the update process on the Command Center and the Sensors; as the TOE does not support automatic updates. The TOE automatically verifies the digital signature of the software update prior to installation and if the digital signature is found to be invalid, the update is not installed. The current executing version of the TOE software is displayed upon login. The TOE implements a self-testing mechanism that is automatically executed upon startup. The TOE provides its own time via the underlying OS's internal clock and a Security Administrator has the ability to manually set the time.

2.5.7 TOE Access

The TOE displays a configurable warning banner prior to user authentication. Remote and local sessions are terminated after an administrator-configurable time period of inactivity. Users are allowed to terminate their own interactive session. Once a session has been terminated, the TOE requires the user to re-authenticate to establish a new session.

2.5.8 Trusted Path/Channels

Security Administrators can remotely manage the Command Center through an SSH channel to access the CLI or HTTPS to access the Web GUI. The Command Center uses a SSH connection to the audit server for remote audit storage.

Security administrators can remotely manage the Sensor through an SSH channel to access the CLI. The Sensor communicates with the Command Center via secure TLS channels.

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 November 29, 2024.

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 November 29, 2024.

3.4 PP Claims

This ST claims exact conformance to the following Protection Profiles:

- Collaborative Protection Profile for Network Devices Version 2.2e (NDcPP), March 23, 2020

3.5 Package Claims

The TOE claims exact compliance to the Collaborative Protection Profile for Network Devices Version 2.2e, which is conformant with CC Part 3.

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

- FAU_GEN_EXT.1
- FAU_STG_EXT.4
- FCS_HTTPS_EXT.1
- FCS_SSHC_EXT.1
- FCS_SSHS_EXT.1
- FCS_TLSC_EXT.1
- FCS_TLSS_EXT.1
- FIA_X509_EXT.2
- FIA_X509_EXT.3
- FMT_MTD.1/CryptoKeys

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

- FIA_X509_EXT.1/ITT
- FPT_ITT.1.1
- FCO_CPC_EXT.1

The PP specifically indicates optional SFRS as allowable options. Therefore, the notion of exact conformance is not violated when not all optional SFRs are claimed. The PP provides both the ST author and evaluation laboratory with instructions on how these claims are to be documented and evaluated.

3.6 Package Name Conformant or Package Name Augmented

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

3.7 Conformance Claim Rationale

Section 1.2 of the NDcPP 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 NDcPP says that example devices that fit this definition include "physical and virtualized routers, firewalls, VPN gateways, IDSs, and switches."

The TOE is a network device which is composed of one Command Center and one or more Sensors. These are two distinct components that must work together for the TOE to perform its intended purpose making this a distributed TOE. When the Forescout eyeInspect product is connected to the network it enables visibility and an understanding of the overall security posture for ICS/SCADA networks.

Therefore, the classification of the TOE being a network device is justified and appropriate as Forescout eyeInspect is an appliance that provides an infrastructure role.

3.8 Technical Decisions

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

The following list of the NDcPP Technical Decisions apply or do not apply to the TOE:

TD #	Title	References	Changes			Analysis to this evaluation	
			SFR	AA	Notes	N/A	Reason
TD0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	FIA_X509_EXT.1/REV, FIA_X509_EXT.1/ITT		X			AA: Testing Update. No ST updates required.
TD0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	FCS_NTP_EXT.1.4, NDSD v2.2		X		X	AA: Testing Update. N/A: SFR not claimed.
TD0536	NIT Technical Decision for Update Verification Inconsistency	AGD_OPE.1, ND SDv2.2		X			AA: Guidance Update. No ST updates required.
TD0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	FIA_X509_EXT.2.2			X		Clarification of an application note. No ST updated required.
TD0546	NIT Technical Decision for DTLS - clarification of Application Note 63	FCS_DTLSC_EXT.1.1			X	X	N/A: SFR not claimed
TD0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	AVA_VAN, ND SDv2.2		X			Clarification of AVA_VAN No ST updates required.

TD0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	ND SDv2.2, FCS_TLSS_EXT.1.4, Test 3		X			AA: Testing Update. No ST updates required.
TD0556	NIT Technical Decision for RFC 5077 question	ND SDv2.2, FCS_TLSS_EXT.1.4, Test 3		X			AA: Testing Update.
TD0563	NiT Technical Decision for Clarification of audit date information	NDcPPv2.2e, FAU_GEN.1.2			X		Clarified date time stamp requirements No ST updates required.
TD0564	NiT Technical Decision for Vulnerability Analysis Search Criteria	ND SDv2.2, AVA_VAN.1			X		Clarified AVA public search requirements.
TD0569	NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	ND SD v2.2, FCS_DTLSS_EXT.1.7, FCS_TLSS_EXT.1.4		X	X		AA: TSS, AGD, ATE Neither session tickets nor resumption are claimed.
TD0570	NiT Technical Decision for Clarification about FIA_AFL.1	FIA_AFL.1			X		Makes FIA_AFL.1 mandatory. FIA_AFL.1 was already claimed. Not marked with footnote as no SFR wording changes were mandated.
TD0571	NiT Technical Decision for Guidance on how to handle FIA_AFL.1	FIA_UAU.1, FIA_PMG_EXT.1			X		Makes FIA_PMG_EXT.1, FIA_AFL.1, and FMT_SMF.1 mandatory. All were previously claimed. Not marked with footnote as no SFR wording changes were mandated.
TD0572	NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	FTP_ITC.1			X		Clarification; no changes to AA or ST required.
TD0580	NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	FCS_CKM.1.1, FCS_CKM.2.1	X	X	X		AA:TSS, Test Footnote 3
TD0581	NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	FCS_CKM.2	X				SFR wording change Footnote 2
TD0591	NIT Technical Decision for Virtual TOEs and hypervisors	A.LIMITED_FUNCTIONALITY, ACRONYMS			X		Assumption wording change. Footnote 1.
TD0592	NIT Technical Decision for Local Storage of Audit Records	FAU_STG			X		Clarification of PP text.
TD0631	NIT Technical Decision for Clarification of public key authentication for SSH Server	ND SDv2.2, FCS_SSHS_EXT.1, FMT_SMF.1	X	X	X		AA:TSS, Testing Update. Footnote 5 and 6

TD0632	NIT Technical Decision for Consistency with Time Data for vNDs	ND SD2.2, FPT_STM_EXT.1.2	X			X	N/A: TOE is not a vND
TD0635	NIT Technical Decision for TLS Server and Key Agreement Parameters	FCS_TLSS_EXT.1.3, NDSD v2.2		X			AA: TSS.
TD0636	NIT Technical Decision for Clarification of Public Key User Authentication for SSH	ND SD2.2, FCS_SSHC_EXT.1	X	X	X		AA: TSS, AGD, Test Footnote 4
TD0638	NIT Technical Decision for Key Pair Generation for Authentication	ND SDv2.2, FCS_CKM.1			X		Requires selection of all key generation schemes needed for FTP_ITC.1, FTP_TRP.1/Admin, FTP_TRP.1/Join, and FPT_ITT.1 in this SFR
TD0639	NIT Technical Decision for Clarification for NTP MAC Keys	FCS_NTP_EXT.1.2, FAU_GEN.1, FCS_CKM.4, FPT_SKP_EXT.1			X	X	N/A: The TOE is not claiming NTP usage
TD0670	NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	ND SD2.2, FCS_TLSC_EXT.2.1	X	X		X	N/A: SFR not claimed
TD0738	NIT Technical Decision for Link to Allowed-With List	Chapter 2			X		PP claimed but note change has no impact on ST.
TD0790	NIT Technical Decision: Clarification Required for testing IPv6	FCS_DTLSC_EXT.1.2, FCS_TLSC_EXT1.2, CPP_ND_V2.2-SD		X			AA: Testing update
TD0792	NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	FIA_PMG_EXT.1, CPP_ND_V2.2-SD		X			AA: TSS
TD0800	Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	FCS_IPSEC_EXT.1.7, FCS_IPSEC_EXT.1.8, CPP_ND_V2.2-SD		X		X	AA: Guidance and testing. N/A: Not claiming IPSEC

Table 9: Technical Decisions

4 Security Problem Definition

4.1 Threats

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

Threat	Threat Definition
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the

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.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic and may allow them to take advantage of any trust relationships with other Network Devices.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

Table 10: 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 NDcPP.

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

Table 11: 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 NDcPP.

Assumption	Assumption Definition
A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security 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[1]	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

[1] TD0591

Assumption	Assumption Definition
	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 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.COMPONENTS_RUNNING	For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Table 12: 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 NDcPP 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 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.COMPONENTS_RUNNING	For distributed TOEs, the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

Table 13: TOE Operational Environment Objectives

4.5 Security Problem Definition Rationale

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

5 Extended Components Definition

5.1 Extended Security Functional Requirements

The extended Security Functional Requirements that are claimed in this ST are taken directly from the PP to which the ST and TOE claim conformance. These extended components are formally defined in the PP in which their usage is required.

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. Note that conversion of British English spelling to American English spelling is not marked as a refinement (e.g., 'authorisation' changed to 'authorization').
- **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 PP (such as if the PP's instantiation of the SFR has a refinement or a completed assignment), the formatting is not preserved. This is so that the reader can identify the operations that are performed by the ST author as opposed to the PP author.

6.2 Security Functional Requirements Summary

The following table lists the SFRs claimed by the TOE:

Class Name	Component Identification	Component Name
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
	FAU_GEN_EXT.1	Security Audit Data Generation for Distributed TOE component
	FAU_STG_EXT.1	Protected Audit Event Storage
	FAU_STG_EXT.4	Protected Local Audit Event Storage for Distributed TOEs
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_HTTPS_EXT.1	HTTPS Protocol
	FCS_RBG_EXT.1	Random Bit Generation
	FCS_SSHC_EXT.1	SSH Client Protocol
	FCS_SSHS_EXT.1	SSH Server Protocol
	FCS_TLSC_EXT.1	TLS Client Protocol Without Mutual Authentication
	FCS_TLSS_EXT.1	TLS Server Protocol Without Mutual Authentication
Communication	FCO_CPC_EXT.1	Component Registration Channel Definition
	FIA_AFL.1	Authentication Failure Management

Identification and Authentication	FIA_PMG_EXT.1	Password Management
	FIA_UAU.7	Protected Authentication Feedback
	FIA_UAU_EXT.2	Password-based Authentication Mechanism
	FIA_UIA_EXT.1	User Identification and Authentication
	FIA_X509_EXT.1/ITT	X.509 Certificate Validation
	FIA_X509_EXT.2	X.509 Certificate Requests
	FIA_X509_EXT.3	X.509 Certificate Requests
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_ITT.1	Basic internal TSF data transfer protection
	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 Banner
Trusted Path /Channels	FTP_ITC.1	Inter-TSF trusted channel
	FTP_TRP.1/Admin	Trusted Path

Table 14: Security Functional Requirements for the TOE

6.3 Security Functional Requirements

6.3.1 Class FAU: Security Audit

6.3.1.1 *FAU_GEN.1 Audit Data Generation*

FAU_GEN.1.1

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

a) Start-up and shut-down of the audit functions;

b) All auditable events for the not specified level of audit; and

c) All administrative actions comprising:

- Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- [no other actions]

d) Specifically defined auditable events listed in Table **15**.

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 **15**.

Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_GEN_EXT.1	None.	None.
FAU_STG_EXT.1	None.	None.
FAU_STG_EXT.4	None.	None.
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_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_RBG_EXT.1	None.	None.
FCS_SSHC_EXT.1	Failure to establish an SSH session.	Reason for failure.
FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
FCS_TLSC_EXT.1	Failure to establish a TLS session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS session	Reason for failure
FCO_CPC_EXT.1	• Enabling communications between a pair of components. • Disabling communications between a pair of components.	Identities of the endpoint pairs enabled or disabled.
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_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_X509_EXT.1/ITT	• 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_ITT.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FPT_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.
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.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failures of the trusted path functions.	None.

Table 15: 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_GEN_EXT.1 Security Audit Data Generation for Distributed TOE component*

FAU_GEN_EXT.1.1

The TSF shall be able to generate audit records for each TOE component. The audit records generated by the TSF of each TOE component shall include the subset of security relevant audit events which can occur on the TOE component.

6.3.1.4 *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 be a distributed TOE that stores audit data on the following TOE components: [*Command Center, Sensor*]

]

FAU_STG_EXT.1.3

The TSF shall [[*rotate compressed audit archived audit files on a First in First out (FIFO) basis according to the following rule:*

- *Delete oldest archived log file*
- *rotate remaining archived log files*
- *close, compress, and archive current log file*
- *open new audit log file to receive current entries*

]]

when the local storage space for audit data is full.

6.3.1.5 *FAU_STG_EXT.4 Protected Local Audit Event Storage for Distributed TOEs*

FAU_STG_EXT.4.1

The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: [

TOE Component	*Behavior when Local Audit Data is Full*
Command Center	[*[Behaves as specified in FAU_STG_EXT.1.3*
Sensor	*Behaves as specified in FAU_STG_EXT.1.3]*]

].

6.3.2 Class FCS: Cryptographic Support

6.3.2.1 *FCS_CKM.1 Cryptographic Key Generation*

FCS_CKM.1.1

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

- ECC schemes using 'NIST curves' [P-256] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;
- FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526].

].

6.3.2.2 *FCS_CKM.2 Cryptographic Key Establishment*

FCS_CKM.2.1[2,3]

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

- 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";
- FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [groups listed in RFC 3526].

].

6.3.2.3 *FCS_CKM.4 Cryptographic Key Destruction*

FCS_CKM.4.1

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

- For plaintext keys in volatile storage, the destruction shall be executed by a [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 [
  - instructs a part of the TSF to destroy the abstraction that represents the key]

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 [CTR, GCM] mode and cryptographic key sizes [256 bits] that meet the following: AES as specified in ISO 18033-3, [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

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [*2048 bits*]
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256 bits*]

]

[2] **TD0581**
[3] **TD0580**

that meet the following: [

- For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3
- For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-384, P-521]; ISO/IEC 14888-3, Section 6.4

].

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

FCS_COP.1.1/Hash

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

6.3.2.8 *FCS_HTTPS_EXT.1 HTTPS Protocol*

FCS_HTTPS_EXT.1.1

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

FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3

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

6.3.2.9 *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 (any), CTR_DRBG (AES)].

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[3 for Sensor, 4 for Command Center]* 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.

6.3.2.10 *FCS_SSHC_EXT.1 SSH Client Protocol*

FCS_SSHC_EXT.1.1

The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [4256, 4344, 5656, 6668, 8268].

FCS_SSHC_EXT.1.2[4]

The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, [no other method].

FCS_SSHC_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [*32,768*] bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.4

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

FCS_SSHC_EXT.1.5

The TSF shall ensure that the SSH public-key based authentication implementation uses [ecdsa-sha2-nistp521] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHC_EXT.1.6

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

FCS_SSHC_EXT.1.7

The TSF shall ensure that [ecdh-sha2-nistp256] and [diffie-hellman-group16-sha512] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.8

The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

FCS_SSHC_EXT.1.9

The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key and [no other methods] as described in RFC 4251 section 4.1.

[4] TD0636

6.3.2.11 *FCS_SSHS_EXT.1 SSH Server Protocol*

FCS_SSHS_EXT.1.1

The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [4256, 4344, 5656, 6668, 8268].

FCS_SSHS_EXT.1.2[5]

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

FCS_SSHS_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [*32,768*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4

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

FCS_SSHS_EXT.1.5

The TSF shall ensure that the SSH public-key based authentication implementation uses [ecdsa-sha2-nistp384, ecdsa-sha2-nistp521] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6

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

FCS_SSHS_EXT.1.7

The TSF shall ensure that [ecdh-sha2-nistp256] and [diffie-hellman-group16-sha512] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8

The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

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

FCS_TLSC_EXT.1.1

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

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

[5] TD0631

]

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, IPv4 address in CN or SAN, and no other attribute types].

FCS_TLSC_EXT.1.3

When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [

- Not implement any administrator override mechanism

].

FCS_TLSC_EXT.1.4

The TSF shall [present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: [secp256r1] and no other curves/groups] in the Client Hello.

6.3.2.13 *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_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

]
and no other ciphersuites.

FCS_TLSS_EXT.1.2

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

FCS_TLSS_EXT.1.3

The TSF shall perform key establishment for TLS using [ECDHE curves [secp256r1] and no other curves].

FCS_TLSS_EXT.1.4

The TSF shall support [no session resumption or session tickets].

6.3.3 Class FCO: Communication

6.3.3.1 *FCO_CPC_EXT.1 Component Registration Channel Definition*

FCO_CPC_EXT.1.1

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

FCO_CPC_EXT.1.2

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

- No channel]

for at least TSF data.

FCO_CPC_EXT.1.3

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

6.3.4 Class FIA: Identification and Authentication

6.3.4.1 *FIA_AFL.1 Authentication Failure Management*

FIA_AFL.1.1

The TSF shall detect when an Administrator configurable positive integer within [*1-3*] 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]* 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].

6.3.4.2 *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 [*8*] and [*60*] characters.

6.3.4.3 *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.

6.3.4.4 *FIA_UAU_EXT.2 Password-based Authentication Mechanism*

FIA_UAU_EXT.2.1

The TSF shall provide a local [password-based] authentication mechanism to perform local administrative user authentication.

6.3.4.5 *FIA_UIA_EXT.1 User Identification and Authentication*

FIA_UIA_EXT.1.1

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

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

FIA_UIA_EXT.1.2

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

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

FIA_X509_EXT.1.1/ITT

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of two 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 [no revocation method].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

FIA_X509_EXT.1.2/ITT

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

6.3.4.7 *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].

FIA_X509_EXT.2.2

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

6.3.4.8 *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].

FIA_X509_EXT.3.2

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

6.3.5 Class FMT: Security Management

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

FMT_MOF.1.1/ManualUpdate

The TSF shall restrict the ability to enable the functions to perform manual updates to Security Administrators.

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

FMT_MTD.1.1/CoreData

The TSF shall restrict the ability to manage the TSF data to Security Administrators.

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

FMT_MTD.1.1/CryptoKeys

The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

6.3.5.4 *FMT_SMF.1 Specification of Management Functions*

FMT_SMF.1.1[6]

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

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [
  - Ability to manage the cryptographic keys;

[6] TD0631

- Ability to configure thresholds for SSH rekeying;
- Ability to configure the interaction between TOE components;
- 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 import X.509v3 certificates to the TOE's trust store;
- Ability to manage the trusted public keys database].

6.3.5.5 *FMT_SMR.2 Restrictions on Security Roles*

FMT_SMR.2.1

The TSF shall maintain the roles:

- Security Administrator.

FMT_SMR.2.2

The TSF shall be able to associate users with roles.

FMT_SMR.2.3

The TSF shall ensure that the conditions

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

are satisfied.

6.3.6 Class FPT: Protection of the TSF

6.3.6.1 *FPT_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.

6.3.6.2 *FPT_ITT.1 Basic internal TSF data transfer protection*

FPT_ITT.1.1

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

6.3.6.3 *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.

6.3.6.4 *FPT_STM_EXT.1 Reliable Time Stamps*

FPT_STM_EXT.1.1

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

FPT_STM_EXT.1.2

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

6.3.6.5 *FPT_TST_EXT.1 TSF Testing*

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests [during initial start-up (on power on), at the request of the authorized user] to demonstrate the correct operation of the TSF: [*specifically defined in Table 16*].

#	Component	Validation
1.	Command Center & Sensor	Standard Linux Filesystem Check
2.	Command Center & Sensor	Hardware Check
3.	Command Center & Sensor	Forescout GPG key is loaded & signature of POST 'truth' file is correct
4.	Command Center & Sensor	Ubuntu LTS version is as expected
5.	Command Center & Sensor	Validate integrity of cryptographic modules
6.	Command Center & Sensor	OS-level dependencies of component are installed

Table 16: Self-Test List

6.3.6.6 *FPT_TUD_EXT.1 Trusted Update*

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a [digital signature] prior to installing those updates.

6.3.7 Class FTA: TOE Access

6.3.7.1 *FTA_SSL_EXT.1 TSF-initiated Session Locking*

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [

- terminate the session]

after a Security Administrator-specified time period of inactivity.

6.3.7.2 *FTA_SSL.3 TSF-initiated Termination*

FTA_SSL.3.1

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

6.3.7.3 *FTA_SSL.4 User-initiated Termination*

FTA_SSL.4.1

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

6.3.7.4 *FTA_TAB.1 Default TOE Access 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.

6.3.8 Class FTP: Trusted Path/Channels

6.3.8.1 *FTP_ITC.1 Inter-TSF trusted channel*

FTP_ITC.1.1

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

FTP_ITC.1.2

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

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for [*send audit data*].

6.3.8.2 *FTP_TRP.1/Admin Trusted Path*

FTP_TRP.1.1/Admin

The TSF shall be capable of using [SSH, HTTPS] 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 summarize 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.

The TOE is classified as a Network Device with a minimum configuration of a Command Center and at least one Sensor. While the two components can function on their own, both components must logically work together in order to perform the product's intended function making this a distributed TOE. The security functionality of the Sensors is not hindered when multiple instantiations are deployed since all Sensors maintain equivalent security features. Therefore, additional Sensors and can be added into a deployment of the TOE without affecting the validity of the functional claims of this ST and the Common Criteria Certification.

The following table defines which distributed TOE component(s) perform the capabilities described by the SFR.

Requirement	Command Center	Sensors
FAU_GEN.1	X	X
FAU_GEN.2	X	X
FAU_GEN_EXT.1	X	X
FAU_STG_EXT.1	X	X
FAU_STG_EXT.4	X	X
FCS_CKM.1	X	X
FCS_CKM.2	X	X
FCS_CKM.4	X	X
FCS_COP.1/DataEncryption	X	X
FCS_COP.1/SigGen	X	X
FCS_COP.1/Hash	X	X
FCS_COP.1/KeyedHash	X	X
FCS_HTTPS_EXT.1	X	
FCS_RBG_EXT.1	X	X
FCS_SSHC_EXT.1	X	
FCS_SSHS_EXT.1	X	X
FCS_TLSC_EXT.1		X
FCS_TLSS_EXT.1	X	
FCO_CPC_EXT.1	X	X
FIA_AFL.1	X	X
FIA_PMG_EXT.1	X	X
FIA_UAU.7	X	X
FIA_UAU_EXT.2	X	X
FIA_UIA_EXT.1	X	X
FIA_X509_EXT.1/ITT		X
FIA_X509_EXT.2		X
FIA_X509_EXT.3	X	
FMT_MOF.1/ManualUpdate	X	X
FMT_MTD.1/CoreData	X	X
FMT_MTD.1/CryptoKeys	X	X
FMT_SMF.1	X	X

FMT_SMR.2	X	X
FPT_APW_EXT.1	X	X
FPT_ITT.1	X	X
FPT_SKP_EXT.1	X	X
FPT_STM_EXT.1	X	X
FPT_TST_EXT.1	X	X
FPT_TUD_EXT.1	X	X
FTA_SSL_EXT.1	X	X
FTA_SSL.3	X	X
FTA_SSL.4	X	X
FTA_TAB.1	X	X
FTP_ITC.1	X	
FTP_TRP.1/Admin	X	X

Table 17: SFR and TOE Component Mapping

8.1 Security Audit

8.1.1 FAU_GEN.1 and FAU_GEN.2

The TOE has the ability to automatically generate audit records based on the events that occur within the TSF. Both the Command Center and the Sensor generate audit records for all administrative functions that occur within each component individually, including the start-up and shut-down of the audit functions, Login/Logout, TSF configuration changes, resetting of passwords, managing cryptographic keys, and all events defined in Table 18. Additionally, Table 18 identifies the audit records that are applicable to the TOE as well as the TOE component which generates them. The local audit function for both the Command Center and the Sensor initiates automatically upon startup of the TOE component.

For both the Command Center and the Sensor, each auditable event generated is associated with the identity of the user that caused the event. The Command Center and the Sensor record the date and time of the event, type of event, subject identity, and the event outcome in the audit records generated by each component respectively. Additional items that are recorded for specified auditable events are listed in Table 18. Any audit records regarding import, deletion, or generation of cryptographic keys contain an identifier, such as reference name to the key, within the audit record generated. For a full list of the audit events samples that are generated by the TOE, please refer to the Supplemental Administrative Guidance Document (AGD).

Requirement	Command Center	Sensors	**Auditable Events**	**Additional Audit Record Contents**
FAU_GEN.1	none	none	none	none
FAU_GEN.2	none	none	none	none
FAU_GEN_EXT.1	none	none	none	none
FAU_STG_EXT.1	none	none	none	none
FAU_STG_EXT.4	none	none	none	none
FCS_CKM.1	none	none	none	none

FCS_CKM.2	none	none	none	none
FCS_CKM.4	none	none	none	none
FCS_COP.1/ DataEncryption	none	none	none	none
FCS_COP.1/ SigGen	none	none	none	none
FCS_COP.1/ Hash	none	none	none	none
FCS_COP.1/ KeyedHash	none	none	none	none
FCS_HTTPS_EXT.1	X		Failure to establish an HTTPS session	Reason for failure
FCS_RBG_EXT.1	none	none	none	none
FCS_SSHC_EXT.1	X		Failure to establish an SSH session	Reason for failure
FCS_SSHS_EXT.1	X	X	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.1		X	Failure to establish a TLS session	Reason for failure
FCS_TLSS_EXT.1	X		Failure to establish a TLS session	Reason for failure
FCO_CPC_EXT.1	X		Enabling communications between a pair of components.	Identities of the endpoint pairs enabled or disabled.
	X		Disabling communications between a pair of components.	Identities of the endpoint pairs enabled or disabled.
FIA_AFL.1	X	X	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	none	none	none	none
FIA_UAU.7	none	none	none	none
FIA_UAU_EXT.2	X	X	All use of the identification and authentication mechanism	Origin of the attempt (e.g., IP address).
FIA_UIA_EXT.1	X	X	All use of the identification and authentication mechanism	Origin of the attempt (e.g., IP address).
FIA_X509_EXT.1/ ITT		X	Unsuccessful attempt to validate a certificate	Reason for failure of certificate validation
		X	Any addition, replacement or removal of trust anchors in the TOE's trust store	Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	none	none	none	none
FIA_X509_EXT.3	none	none	none	none

FMT_MOF.1/ ManualUpdate	X	X	Any attempt to initiate a manual update	none
FMT_MTD.1/ CoreData	none	none	none	none
FMT_MTD.1/ CryptoKeys	none	none	none	none
FMT_SMF.1	X	X	All management activities of TSF data.	none
FMT_SMR.2	none	none	none	none
FPT_APW_EXT.1	none	none	none	none
FPT_ITT.1	X	X	Initiation of the trusted channel.	Identification of the initiator and target of failed trusted channels establishment attempt.
	X	X	Termination of the trusted channel.	Identification of the initiator and target of failed trusted channels establishment attempt.
	X	X	Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FPT_SKP_EXT.1	none	none	none	none
FPT_STM.1	X	X	Discontinuous changes to time - either Administrator actuated or changed via an automated process.	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TUD_EXT.1	X	X	Initiation of update	none
	X	X	Result of the update attempt (success or failure)	none
FTA_SSL_EXT.1	X	X	The termination of a local session by the session locking mechanism. OR The termination of an interactive session.	none

FTA_SSL.3	X	X	The termination of a remote session by the session locking mechanism.	none
FTA_SSL.4	X	X	The termination of an interactive session.	none
FTA_TAB.1	none	none	none	none
FTP_ITC.1	X		Initiation of the trusted channel.	Identification of the initiator and target of failed trusted channels establishment attempt.
	X		Termination of the trusted channel.	Identification of the initiator and target of failed trusted channels establishment attempt.
	X		Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/ Admin	X	X	Initiation of the trusted path.	none
	X	X	Termination of the trusted path.	none
	X	X	Failure of the trusted path functions.	none

Table 18: Auditable Events

8.1.2 FAU_GEN_EXT.1

Both the Command Center and Sensor generate their own auditable events. The auditable events generated by the Command Center and the Sensors are shown above in Table 18. Each Sensor generates its own audits regardless of the number of Sensors in a deployment.

8.1.3 FAU_STG_EXT.1

The distributed TOE is comprised of two components: the Command Center and the Sensor. Each component generates its own audit records based on the events that occur on each individual component respectively. In the evaluated configuration, the generated audit records are first saved locally within the generating TOE component. The TOE's Command Center collects all TOE audit data, and securely transmits the audit data via a SSH channel to the Operational Environment's audit server per its defined configuration.

On the Command Center, the TSF provides the ability for a Security Administrator to configure the forwarding of the audit trail to an external audit server in the Operational Environment. The audit data generated by the Command Center is aggregated in the ei-aggregated.log file for local storage and sending to the audit server. The Command Center determines if the contents of the ei-aggregated.log file need to be sent to the audit server on a time based frequency (default 1 minute) but will only send the contents when the size of the ei-aggregated.log file reaches a specific size (default 2MB). The frequency and file size

values for this process are configurable by the Security Administrator, the default values are deemed to be an acceptable frequency.

A Sensor does not have a direct connection to the audit server. The audit data generated by the Sensor is stored locally (i.e., current.log file) and forwarded automatically in near real-time to the Command Center over TLS. The Command Center buffers audit data from one or more Sensors in the sensors.log file; until the Command Center is ready to send the audit data to the remote audit server. The Command Center determines if the contents of the sensors.log file need to be sent to the audit server on a time based frequency (default 1 minute) but will only send the contents when the size of the sensors.log file reaches a specific size (default 2MB). The frequency and file size values for this process are configurable by the Security Administrator, the default values are deemed to be an acceptable frequency.

In the event that the connection is lost between the Command Center and the audit server:

- The Command Center audit data generated post connection loss, continues to be stored locally on the Command Center.
- Audit data sent by a Sensor to the Command Center, for forwarding to the external audit server, continues to be stored on the Command Center.
- Once the lost connection to the audit server is re-established, the TOE automatically begins forwarding the currently stored Command Center (i.e., ei-aggregated.log file) and Sensor audit (i.e., sensors.log file) records based on the established configuration. Note that during a connection outage, the Command Center will continue to perform its log rotation of these aggregated sources. Therefore, any audit records located in a log file instance that has been deleted due to rotation will not be sent to the remote audit server.

Audit data generated by the Sensor during a connection outage between the Sensor and the Command Center is stored locally on the Sensor, and forwarded to the Command Center once the connection is reestablished. Note that during a connection outage, the Sensor will continue to perform its log rotation of its current.log file. Therefore, any audit records that have been deleted due to its rotation will not be sent to the Command Center.

All audit files perform file rotation which is triggered based upon the rotation cadence listed in Table 19 below. When the file rotation sequence is triggered, the oldest log file is deleted from the archive to free up space for the rotation. The remaining log files are rotated and renamed. The currently open log file is closed, compressed, and archived. Finally, a new audit log file is created to store new entries.

TOE Component	Log file	Rotation cadence	Max number of rotations (current + historical)
Command Center	ei-aggregated.log	Every 2MB	25
Command Center	sensors.log	Every 2MB	25
Sensor	current.log	Every 1GB	1 A new current.log file is created once it becomes full.

Table 19: Application Related Log Files Rotation Rules

The amount of audit data that is stored locally on each TOE component is determined by the maximum file size before a rotation occurs and the number of rotations possible. Therefore, the amount of audit storage for the ei-aggregated.log storage is 60MB. For all log files, when the storage space is exhausted a rotation occurs.

For both the Command Center and the Sensor, all audit records can be reviewed only by the Security Administrator through each component's respective CLI. The Security Administrator has the ability to

view, edit or delete audit records. Additionally, user activity records can be viewed by the Security Administrator using the web UI. The audit data is protected against unauthorized access via the TOE's role-based access control mechanisms via its user interfaces.

8.1.4 FAU_STG_EXT.4

The Command Center and the Sensor comprise a full deployment of the TOE. The audit data generated by each individual TOE component is stored locally in the appropriate location of each component's respective file system. Refer to Table 18 for a full breakdown of auditable events generated per component.

Both the Command Center and the Sensor use a compressed log file rotation scheme to ensure adequate storage space for new audit log files. When the file rotation sequence is triggered, the oldest log file is deleted from the archive to free up space for the rotation. The remaining log files are rotated and renamed. The currently open log file is closed, compressed, and archived. Finally, a new audit log file is created to store new entries. Details regarding log rotation cadence and the maximum number of audit files are listed in section 8.1.3.

8.2 Cryptographic Support

The TOE implements two different cryptographic libraries: Bouncy Castle and OpenSSL. The Command Center utilizes Bouncy Castle for TLS and HTTPS communications, and OpenSSL for SSH communications. The Sensor utilizes OpenSSL for TLS and SSH communication. Both libraries include algorithms that are certified under the following consolidated CAVP certificates:

a) BC-FJA (Bouncy Castle FIPS Java API) Software Version 1.0.2.1 under CAVP Certificate #A6120

b) OpenSSL 3.0.8 FIPS library under CAVP Certificate #A6128

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

SFR	Algorithm/Protocol	CAVP Cert #	
		Bouncy Castle	**OpenSSL**
FCS_CKM.1	ECC using NIST curves P-256, per FIPS PUB 186-4	#A6120 ECDSA KeyGen & KeyVer (FIPS186-4)	#A6128 ECDSA KeyGen & KeyVer (FIPS186-4)
	FFC using safe-prime groups NIST Special Publication 800-56A Revision 3 and RFC 3526.	N/A – Bouncy Castle does not provide FFC services	#A6128 Safe Primes KeyGen & KeyVer MODP-2048
FCS_CKM.2	Elliptic curve-based key establishment NIST Special Publication 800-56A Revision 3	#A6120 KAS-ECC-SSC Sp800-56Ar3	#A6128 KAS-ECC-SSC Sp800-56Ar3
	FFC using safe-prime groups NIST Special Publication 800-56A Revision 3 and groups listed in RFC 3526.	N/A – Bouncy Castle does not provide FFC services	#A6128 KAS-FFC-SSC Sp800-56Ar3
FCS_COP.1/ DataEncryption	AES GCM 256 bits	#A6120 AES-GCM	N/A
	AES CTR 256 bits	N/A	#A6128 AES-CTR

FCS_COP.1/ SigGen	RSA FIPS 186-4 Signature Services 2048 bits	#A6120 RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	N/A
	ECDSA FIPS 186-4 Signature Services 256 bits	N/A	#A6128 ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)
FCS_COP.1/ Hash	SHA-256, SHA-384, and SHA-512	#A6120 SHA2-256, SHA2-384, SHA2-512	N/A
	SHA-256 and SHA-512	N/A	#A6128 SHA2-256, SHA2-512
FCS_COP.1/ KeyedHash	HMAC-SHA-384	#A6120 HMAC-SHA2-384	N/A
	HMAC-SHA-256 and HMAC- SHA-512	N/A	#A6128 HMAC-SHA2-256, HMAC-SHA2-512
FCS_RBG_EXT.1	Hash DRBG	#A6120 Hash_DRBG	N/A
	CTR_DRBG	N/A	#A6128 Counter_DRBG

Table 20: Cryptographic Algorithm Table for Bouncy Castle and OpenSSL on the Command Center

SFR	Algorithm/Protocol	CAVP Cert #
FCS_CKM.1	ECC using NIST curves P-256, per FIPS PUB 186-4	#A6128 ECDSA KeyGen & KeyVer (FIPS186-4)
	FFC using safe-prime groups NIST Special Publication 800-56A Revision 3 and RFC 3526.	#A6128 Safe Primes KeyGen / KeyVer MODP-2048
FCS_CKM.2	Elliptic curve-based key establishment NIST Special Publication 800-56A Revision 3	#A6128 KAS-ECC-SSC Sp800-56Ar3
	FFC using safe-prime groups NIST Special Publication 800-56A Revision 3 and groups listed in RFC 3526.	#A6128 KAS-FFC-SSC Sp800-56Ar3
FCS_COP.1/ DataEncryption	AES CTR 256 bits, AES GCM 256 bits	#A6128 AES-GCM AES-CTR
FCS_COP.1/SigGen	RSA FIPS 186-4 Signature Services 2048 bits	#A6128 RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)
	ECDSA FIPS 186-4 Signature Services 256 bits	#A6128 ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)
FCS_COP.1/Hash	SHA-256, SHA-384, and SHA-512	#A6128 SHA2-256, SHA2-384, SHA2-512

FCS_COP.1/ KeyedHash	HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512	#A6128 HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512
FCS_RBG_EXT.1	CTR_DRBG	#A6128 Counter DRBG

Table 21: Cryptographic Algorithm Table for OpenSSL on the Sensor

8.2.1 FCS_CKM.1

The Command Center generates an ECC key using NIST curve P-256 in accordance with FIPS PUB 186-4 (Digital Signature Standard (DSS) Appendix B.4) supporting a 256-bit key size. The ECC keys are generated in support of device authentication for TLS and SSH.

The Sensor generates an ECC key using NIST curve P-256 in accordance with FIPS PUB 186-4 (Digital Signature Standard (DSS) Appendix B.4) supporting a 256-bit key size. The ECC keys are generated in support of device authentication for SSH. The Sensor is not a HTTPS or TLS server. Therefore, the Sensor does not generate certificates to support TLS communications.

Additionally, both the Command Center and Sensor generate FFC keys in accordance with NIST Special Publication 800-56A Revision 3 and RFC 3526. The FFC keys are generated in support of device authentication for SSH.

The TOE cryptographic implementation is as follows:

- Bouncy Castle provides the TLS Elliptic curve-based key generation services for the Command Center
- OpenSSL provides the SSH Elliptic curve-based and FFC key generation services for the Command Center
- OpenSSL provides the TLS and SSH Elliptic curve-based and FFC key generation services for the Sensor

The TOE's key generation cryptographic implementations are validated under CAVP. See Tables 20 & 21 Cryptographic Algorithm Table for certification numbers.

8.2.2 FCS_CKM.2

The Command Center implements Elliptic curve-based key establishment, that complies with all sections regarding Elliptic curve-based key pair generation and key establishment NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography", for:

- TLS server communications (FCS_TLSS_EXT.1) to establish TLS connections from the administrative workstation for accessing the Web GUI for remote administrative purposes and from the Sensor for TOE-TOE communications
- SSH server communications (FCS_SSHS_EXT.1) to establish SSH connections from the administrative workstation for accessing the CLI for remote administrative purposes
- SSH client communications (FCS_SSHC_EXT.1) to the audit server

The Sensor implements Elliptic curve-based key establishment, that complies with all sections regarding Elliptic curve-based key pair generation and key establishment NIST Special Publication 800-56A Revision

3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”, for:

- TLS client communications (FCS_TLSC_EXT.1) with the Sensor of the Command Center
- SSH server communications (FCS_SSHS_EXT.1) to establish SSH connections from the administrative workstation for accessing the CLI for remote administrative purposes

Both the Command Center and Sensor implement FFC key establishment methods using safe prime groups for SSH server communications (FCS_SSHS_EXT.1) with the Remote Administrative Workstation for accessing the CLI for remote administrative purposes. The implementations comply with all sections regarding FFC based key pair generation and key establishment as defined in NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and groups listed in RFC 3526.

The TOE cryptographic implementation is as follows:

- Bouncy Castle provides the TLS Elliptic curve-based key establishment services for the Command Center
- OpenSSL provides the SSH Elliptic curve-based and FFC key establishment services for the Command Center
- OpenSSL provides the TLS and SSH Elliptic curve-based and FFC key establishment services for the Sensor

The TOE’s key establishment cryptographic implementations are validated under CAVP. See Tables 20 & 21 Cryptographic Algorithm Table for certification numbers.

8.2.3 FCS_CKM.4

The following table identifies the keys and CSPs that are applicable to the TOE as well as the following data regarding the key material: applicable TOE component(s), origin, storage location, and method of destruction. The TOE is not subject to any situations that would prevent or delay key destruction, and strictly conforms to the key destruction requirements.

Key Material Name	TOE Component	Origin	Storage	Zeroization / Destruction
Diffie-Hellman Shared Secret	Command Center/Sensor	Generated by TOE’s SSH Server / SSH Client applications	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00)*. 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	Command Center/Sensor	Generated by TOE’s SSH Server / SSH Client applications	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00)*. The key is zeroized immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Automatically zeroized after DH exchange

Key Material Name	TOE Component	Origin	Storage	Zeroization / Destruction
SSH session keys	Command Center/Sensor	Generated by TOE's SSH Server / SSH Client applications	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00)*. 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	Command Center/Sensor	Generated on TOE component by Security Administrator during initial setup of device.	Filesystem	The Security Administrator destroys this key via the CLI by entering a command that will delete the key. When the TOE processes this command, it destroys the abstraction that represented the key. The Security Administrator would perform this action when they want to replace the key.
X.509 Certificate	Command Center	Generated on Command Center by Security Administrator during initial setup.	Filesystem	The Security Administrator destroys this key via the CLI by entering a command that will delete the key. When the TOE processes this command, it destroys the abstraction that represented the key. The Security Administrator would perform this action when they want to replace the key.
TLS session keys	Command Center/Sensor	Generated by TOE's TLS Server / TLS Client applications	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00)*. The key is zeroized immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Automatic zeroized after TLS session is terminated.

Table 22: Crypto key destruction table

*A ZeroizableSecretKey instance is returned as part of the agreed key algorithm. That instance implements zeroize() function that gets called upon destroy().

8.2.4 FCS_COP.1/DataEncryption

The TOE performs encryption and decryption using the AES algorithm in CTR and GCM modes with key sizes of 256 bits. The AES algorithm meets ISO 18033-3, CTR meets ISO 10116 and GCM meets ISO 19772. The TOE's AES implementation is validated under CAVP. See Tables 20 & 21 Cryptographic Algorithm Table for certification numbers.

The TOE cryptographic implementation is as follows:

- For the Command Center, Bouncy Castle supports: TLS communication: AES-GCM-256
- For the Command Center, OpenSSL supports:
    - SSH communication: AES-CTR-256

  - DRBG: AES-CTR-256
- For the Sensor, OpenSSL supports:
  - TLS communication: AES-GCM-256
  - SSH communication: AES-CTR-256
  - DRBG: AES-CTR-256

The TOE's AES encryption cryptographic implementations are validated under CAVP. See Tables 20 & 21 Cryptographic Algorithm Table for certification numbers.

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. 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 performs signature generation and validation using Elliptic Curve Digital Signature Algorithm (ECDSA). The TOE supports ECDSA with 256-bit key size and implements the NIST P-384 and P-521 curves. The ECDSA implementation meets ISO/IEC 14888-3 Section 6.4 and FIPS PUB 186-4. The TOE's RSA and ECDSA implementations are validated under CAVP. See Tables 20 & 21 Cryptographic Algorithm Table for certification numbers.

The TOE cryptographic implementation is as follows:

- Bouncy Castle provides the RSA signature generation and verification services for TLS communications on the Command Center
- OpenSSL provides the RSA signature generation and verification services for the TLS communications on the Sensor
- OpenSSL provides the ECDSA signature generation and verification services for the SSH communications on the Command Center and Sensor

8.2.6 FCS_COP.1/Hash

The Command Center and Sensor provide cryptographic hashing services using 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 20 & 21 Cryptographic Algorithm Table for certification numbers.

Hash support is applicable to both TOE components and the cryptographic libraries being implemented. The following list identifies the hashing support provided by both TOE components:

- password hashing of all Web GUI user passwords stored on the Command Center (FPT_APW_EXT.1: SHA-256)
- password hashing of all CLI user passwords stored on the Command Center and Sensor (FPT_APW_EXT.1: SHA-512)
- trusted updates digital signature verification on both TOE components (FPT_TUD_EXT.1: SHA-512)
- TSF self-testing hash value check verification on both TOE components (FPT_TST_EXT.1: SHA-256)
- TLS Server communications on the Command Center (FCS_TLSS_EXT.1:SHA-384)

- TLS Client communications on the Sensor (FCS_TLSC_EXT.1:SHA-384)
- SSH Client communications on the Command Center (FCS_SSHC_EXT.1: SHA-256 and SHA-512)
- SSH Server communications on both TOE components (FCS_SSHS_EXT.1: SHA-256 and SHA-512)
- Hash DRBG on the Command Center (FCS_RBG_EXT.1: SHA-512)

The TOE cryptographic implementation is as follows:

- Bouncy Castle provides SHA-256, SHA-384, and SHA-512 hash services for the Command Center
- OpenSSL provides SHA-256 and SHA-512 keyed hash services for the Command Center
- OpenSSL provides SHA-256, SHA-384, and SHA-512 hash services for the Sensor

The TOE's hash cryptographic implementations are validated under CAVP. See Tables 20 & 21 Cryptographic Algorithm Table for certification numbers.

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-256 [key-size: 256 bits, hash function: SHA-256, digest size: 256 bits, block size: 512 bits, MAC lengths: 256 bits] for SSH communication support only
- HMAC-SHA-384 [key-size: 384 bits, hash function: SHA-384, digest size: 384 bits, block size: 1024 bits, MAC lengths: 384 bits] for TLS communication support only
- HMAC-SHA-512 [key-size: 512 bits, hash function: SHA-512, digest size: 512 bits, block size: 1024 bits, MAC lengths: 512 bits] for SSH communication support only

The TOE cryptographic implementation is as follows:

- Bouncy Castle provides HMAC-SHA-384 keyed hash services for the Command Center
- OpenSSL provides HMAC-SHA-256 and HMAC-SHA-512 keyed hash services for the Command Center
- OpenSSL provides HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 keyed hash services for the Sensor

The TOE's HMAC cryptographic implementations are validated under CAVP. See Tables 20 & 21 Cryptographic Algorithm Table for certification numbers.

8.2.8 FCS_HTTPS_EXT.1

All HTTPS connections initiated from the Remote Management Workstation to the Command Center Web GUI are for remote administration of the TOE. The implementation of HTTPS within the TOE complies with RFC 2818 and utilizes a TLS (no mutual authentication) in accordance with the requirements specified in FCS_TLSS_EXT.1. If a peer certificate is presented to the TOE, the TOE does not require client authentication if the peer certificate is deemed invalid. The following summarizes how the TOE conforms to RFC 2818.

The Sensor does not support HTTPS connections.

Section 2.1 Connection Initiation:	The TOE only operates as a HTTPS Server; therefore, this section is not relevant to the TOE.
Section 2.2 Connection Closure:	The TOE sends TLS closure alert when terminating an HTTPS connection. The TOE does not support session reuse. The TOE meets the behavior as described, without deviation.
Section 2.2.1 Client Behavior:	The TOE only operates as a HTTPS Server; therefore, this section is not relevant to the TOE.
Section 2.2.2 Server Behavior:	The TOE does not support session resumption. The TOE attempts to initiate an exchange of closure alerts with the client before closing the connection.
Section 2.3 Port Number:	The TOE utilizes TCP port 443 to listen for incoming HTTPS connections.
Section 2.4 URI Format:	The TOE supports and requires the https:// URI protocol identifier prefix for incoming HTTPS requests.
Section 3.1 Server Identity:	The TOE only operates as a HTTPS Server; therefore, this section is not relevant to the TOE.
Section 3.2 Client Identity:	The TOE does not support mutual authentication for the HTTPS Server interface.

8.2.9 FCS_RBG_EXT.1

The Command Center implements the Hash_DRBG (SHA-512) from Bouncy Castle and the CTR_DRBG (AES-256) from OpenSSL. The Command Center uses Bouncy Castle's Hash_DRBG to support all TLS functionality and OpenSSL's CTR_DRBG to support all SSH functionality.

The Sensor implements the CTR_DRBG (AES-256) from OpenSSL to support all TLS and SSH functionality.

The DRBGs 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 TOE relies on kernel modules to gather and provide entropy for the TOE's random requirements. The Sensor's DRBG uses 3 software-based noise sources (i.e., non-periodic interrupts, human interface device, disk input/output) as stated in the proprietary Entropy Assessment Report. The Command Center's DRBG uses 4 software-based noise sources (i.e., non-periodic interrupts, human interface device, disk input/output, haveged) as stated in the proprietary Entropy Assessment Report. The DRBGs are seeded with a combined minimum of 256-bit security strength.

Both Bouncy Castle and OpenSSL read entropy from /dev/random. The /dev/random entropy pools are protected by being in kernel memory and are not accessible from user space. The entropy source is described in greater detail in the proprietary Entropy Assessment Report.

The TOE's DRBG cryptographic implementations are validated under CAVP. See Tables 20 & 21 Cryptographic Algorithm Table for certification numbers.

8.2.10 FCS_SSHC_EXT.1

The Command Center has SSH client capabilities for initiating an SSH channel to the audit server for exporting audit records. The Command Center acts as a SSHv2 client for remote CLI sessions that complies with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5656, 6668, and 8268. The implementation of SSH supports public key-based authentication only. Per RFC 4251 section 4.1, the TOE's SSH client

implementation will authenticate the identity of the audit server (i.e., SSH server) by using its local database (i.e., ~/.ssh/known_hosts) which associates each host name with its corresponding public key.

The SSH implementation detects for large packets greater than 32,768 bytes and automatically drops the connection accordingly, as described in RFC 4253. Additionally, the TSF enforces that the connection is rekeyed after no longer than one hour, and no more than one gigabyte of transmitted data, whichever threshold is reached first. The SSH rekey time and size threshold parameters are administratively configurable via the CLI. One hour and one gigabyte are the maximum settings allowed for the rekey threshold parameters in the evaluated configuration.

In the evaluated configuration, the TOE's SSHv2 client implementation only supports (rejecting all other algorithms):

- ecdsa-sha2-nistp521 for the client public-key based authentication
- aes256-ctr for its encryption algorithms
- ecdsa-sha2-nistp521 for the host public-key based authentication
- hmac-sha2-256 and hmac-sha2-512 for SSH data integrity MAC algorithm
- ecdh-sha2-nistp256 and diffie-hellman-group16-sha512 for key exchange method in accordance with RFC 3526 Section 3

The TOE cryptographic implementation is as follows:

- OpenSSL provides the SSH cryptographic services for the SSH connection when the Command Center acts as an SSH client for establishing connections between the Command Center and audit server
- The Sensor does not support SSH client functionality

8.2.11 FCS_SSHS_EXT.1

Both the Command Center and the Sensor of the TOE have SSH connection capability. SSH is used to connect securely to the Command Center and the Sensor for remote administration via the CLI. The Command Center and the Sensor act as a SSHv2 server for remote CLI sessions that complies with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5656, 6668, and 8268. The TOE's implementation of SSH supports both public key-based and password-based user authentication. If a public key is presented for user authentication, the TOE verifies that the SSH client's presented public key matches one that is stored within the SSH server's authorized keys database, which establishes the user's identity. If the SSH client's presented public key does not match a stored key on the TOE, the TOE considers this a failed authentication attempt and the connection is not established. For password-based authentication attempts, the presented user credentials are verified using the TOE's native authentication mechanism. If the presented user credentials cannot be verified, then the connection is not established.

The SSH implementation detects all large packets greater than 32,768 bytes and automatically drops the connection accordingly, as described in RFC 4253. Additionally, the TSF enforces that the connection is rekeyed after no longer than one hour, and no more than one gigabyte of transmitted data, whichever threshold is reached first. The SSH rekey time and size threshold parameters are administratively configurable via the CLI. One hour and one gigabyte are the maximum settings allowed for the rekey threshold parameters in the evaluated configuration.

In the evaluated configuration, the TOE's SSHv2 server implementation only supports (rejecting all other algorithms):

- ecdsa-sha2-nistp384 and ecdsa-sha2-nistp521 for the client public-key based authentication
- aes256-ctr for its encryption algorithms
- ecdsa-sha2-nistp384 and ecdsa-sha2-nistp521 for the host public-key based authentication
- hmac-sha2-256 and hmac-sha2-512 for SSH data integrity MAC algorithm
- ecdh-sha2-nistp256 and diffie-hellman-group16-sha512 for key exchange method in accordance with RFC 3526 Section 3

The TOE cryptographic implementation is as follows:

- OpenSSL provides the SSH cryptographic services when the Command Center acts as an SSH server
- OpenSSL provides the SSH cryptographic services when the Sensor acts as an SSH server

8.2.12 FCS_TLSC_EXT.1

The Sensors are configured to support TLS v1.2 only. All other SSL and TLS versions are rejected and the connection is not established with the TLS server. In the evaluated configuration, the Sensor components only present the secp256r1 elliptic curve in the Client Hello and only use the TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 algorithm for establishing connections to the Command Center for inter-TOE communication. This default configuration is defined as part of the installation process and cannot be modified.

The Sensor, upon the presentation of the X.509v3 server host certificate, validates the certificate per FIA_X509_EXT.1/ITT requirements. The Sensor establishes a trusted channel only if the peer certificate from the Command Center is valid. The Sensor does not perform CRL or OCSP for the FPT_ITT.1 defined channel.

In the evaluated configuration, the Sensor only supports Common Name (CN) and Subject Alternative Name (SAN) reference identifiers that are using IPv4 address values, and mandates the presence of the SAN. Canonical formatting according to RFC 3986 is enforced. If the SAN is missing, empty, or does not match the IPv4 reference identifier, the Sensor terminates the connection. Only if the SAN fields entry matches the reference identifier and the certificate is valid, in according to the FIA_X509_EXT.1/ITT requirements, is the connection established. The TOE does not support the use of IPv6 addresses, URI, DNS (FQDN), service name reference identifiers, wildcards or pinned certificates. There is no administrative override mechanism to force the connection if the peer certificate is deemed invalid.

The TSF converts that IP address, obtained from the certificate, from ASN.1 to the binary representation of the textual string of the IP address. The TSF also converts the IP address from the established network connection to the binary representation of the textual string of the IP address. The two representations are then compared to determine what action is performed next.

The TOE cryptographic implementation is as follows:

- OpenSSL provides the same services for the Sensor when acting as a TLS client

There is no TLS Client functionality being claimed for the Command Center.

8.2.13 FCS_TLSS_EXT.1

The Command Center TLS server functionality is configured to support TLS v1.2 only. All other versions of SSL and TLS are rejected and the connection not established. In the evaluated configuration, the Command Center, when acting as a TLS server, is configured to only use TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 algorithm with ECDHE curve secp256r1 for establishing connections from:

- the remote administrator workstation to access Web GUI for remote management
- the Sensors for inter-TOE communication

Session resumption and session tickets are not supported. Mutual authentication is not claimed.

The TOE cryptographic implementation is as follows:

- Bouncy Castle provides the cryptographic services for key establishment and encryption TLS channel when the Command Center acts as a TLS server
- The Sensor does not support TLS Server functionality

8.3 Communication

8.3.1 FCO_CPC_EXT.1

The Command Center and the Sensors utilize TLS v1.2 to maintain a secure connection and communication between TOE components. There is no automated registration process or separate channel used during the registration process. All configuration and approval is done locally at each individual TOE component by a user assigned the Security Administrator privileges.

Before a connection can be established between the Command Center and a Sensor, a Security Administrator must authenticate to the Sensor's CLI and assume the *silentdefense* role. The Security Administrator loads the appropriate software, imports the Command Center public certificate, and define the Command Center's IP address in the Sensor's configuration. After the IP address has been specified, the Sensor reaches out to the Command Center by establishing a TLS connection.

From the Command Center's Web GUI, an authenticated user with the Admin or the Analyst role, must enable the communication initiated by the Sensor (identified by its IP address) before any more information can be exchanged between the TOE components. Once the enablement action has been performed, the same TLS connection is used for all TOE communication. Through the Command Center's Web GUI, the Admin and the Analyst roles also have the Security Administrator privileges to disable the inter-TOE communication between a Sensor and the Command Center by deleting the Sensor from configuration. Once the Sensor is deleted from the Command Center's configuration, the Sensor is no longer considered a TOE component and goes back to the state of attempting to enroll with the Command Center. Until the enablement step is performed again, no TOE information will be exchanged between the Command Center and Sensor.

8.4 Identification and Authentication

8.4.1 FIA_AFL.1

The TSF provides a configurable counter for consecutive failed remote authentication attempts by a user account. The TSF locks that user account once the configured failure counter threshold is reached.

Credentials for all user accounts are distinctly configured for each of the TOE component's accessible interfaces. Credentials for one TOE interface cannot be used to access another TOE interface (i.e., Command Center Web GUI credentials do not grant access to the Command Center's nor Sensor's remote CLI). The only instance where user credentials from one remote interface would grant access to another remote interface would be if the usernames and passwords were purposefully configured to be the same. A valid login, that happens prior to the failure counter reaching its threshold, resets the counter to zero for that user account.

For the Command Center's Web GUI, the Security Administrator possesses the capability to change the maximum number of failed authentication attempts before a user account is temporarily locked through the Web GUI. The default number of consecutive failed attempts is set to 3. After a user has failed to authenticate within the configured number of attempts, the user account is locked for 15 minutes. Once the lock interval has passed, the account is automatically unlocked and the user can attempt to remotely authenticate.

The Command Center remote CLI and the Sensor remote CLI lockout configuration must be set by a Security Administrator using the CLI. The default number of consecutive failed user login attempts is set to 3. The Security Administrator may choose to set/enable the automatic unlock time feature by entering the time in seconds or disable the automatic unlock time feature by setting the unlock time to 0. A 0 forces the need for a Security Administrator to manually unlock the locked account.

In order to prevent a situation where no Security Administrator can log into the TOE, the local CLI user *silentdefense* is not subject to lockout feature. The *silentdefense* user cannot log into the TOE via SSH. In order to access the CLI remotely, a user must SSH into the Command Center or Sensor using the *eyeInspect* user. The *eyeInspect* user is subject to the locking mechanism and cannot log into the TOE locally. Once the *eyeInspect* user has successfully logged into the TOE a second password is required in order to escalate privileges to the *silentdefense* user and have Security Administrator rights. If the *eyeInspect* user account were to lock, the Security Administrator using the CLI must manually unlock the account by executing the following command:

```
sudo faillock --user eyeinspect --reset
```

8.4.2 FIA_PMG_EXT.1

A Security Administrator can configure the Web GUI password length between 8 and 60 characters using the Web GUI. The minimum password length for the CLI must be configured by the Security Administrator using the CLI on each TOE component. For the CLI interface, it is recommended to set the password length between 8 and 60 characters. Passwords can be composed of any combination of upper and lower-case letters, numbers and special characters. The accepted special characters include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")".

8.4.3 FIA_UAU.7

When authenticating to the Command Center and the Sensor via the local CLI, no characters are shown as the password is typed. 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.

8.4.4 FIA_UAU_EXT.2 and FIA_UIA_EXT.1

No administrative actions can be performed on any of the TOE components prior to a Security Administrator's successful authentication. The displaying of the pre-authentication warning banner is the only TOE functionality that is available to an unauthenticated user, regardless of TOE component and authentication interface. Access is only granted once the user provided authentication credentials are validated using the method of authentication assigned to that interface.

When connecting to the Command Center via the Web GUI, which establishes a HTTPS connection, the Command Center Login page displays the pre-authentication warning banner. To authenticate to the Command Center the user must provide username and password credentials which are validated against the Command Center's native authentication mechanism.

When connecting to the Command Center or Sensor using an SSH client to gain access to each component's respective CLI (SSH CLI), the TOE displays the pre-authentication warning banner. Security Administrator's authenticate to the Command Center or Sensor using username and password credentials which are validated against the individual component's native authentication mechanism. Additionally, the Command Center and the Sensor support public key-based authentication for SSH connections.

When connecting to the Command Center and the Sensor locally via the component's respective Terminal (local CLI), a pre-authentication warning banner is displayed. Security Administrator's authenticate to the Command Center or Sensor using username and password credentials which are validated against the individual component's native authentication mechanism.

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

The TOE utilizes X.509v3 certificates to support the establishment of TLS connections between the Command Center and the Sensor in accordance with RFC 5280. In order for the Sensor to component to establish a TLS connection to the Command Center, the Command Center's trusted X.509v3 CA root certificate must be imported into the Sensor trust store.

The Sensor immediately performs X.509v3 certificate validity checking upon the Command Center presenting its certificate during a TLS connection request. For this connection, the Command Center presents a certificate with a minimum path length of 2 certificates. The Sensor 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.

- The TSF validates a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF validates the extendedKeyUsage field according to the following rules:
    - Server certificates presented for TLS must have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.

Note that the following extendedKeyUsage field rules are not supported by the TOE:

- Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
- OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

The Sensor does not perform revocation status checking of the Command Center's server certificate. As no revocation checking is performed for the distributed TOE interfaces, revocation checking is trivially satisfied for these interfaces and the Sensor accepts the certificate as long as the certificate is valid according to all other rules. Once the Command Center's certificate has been successfully validated the connection will be established.

An administrator for the Command Center can generate a Certificate Request as specified in RFC 2986 containing the public key and "Common Name" in order for the Command Center to have its own certificate. The chain of certificates is validated from the root CA when the CA Certificate Response is received. For connections from the Sensor to the Command Center, the Command Center must have a server X.509v3 certificate, with a minimum path length of two, which is presented to the Sensor.

8.5 Security Management

8.5.1 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 includes restrictions for these roles and functions.

The TOE utilizes role-based access control (RBAC), as described in FMT_SMR.2, to restrict access to the administrative functions that manage the TSF data. Display of the pre-authentication warning banner is the only TOE functionality available prior to identification and authentication. The TOE limits the presented functionality based on the privileges bound to the authenticated user. The available functionality presented to an authenticated user is based on the group of permissions and the privileges associated with the permissions aligned to the authenticated user's assigned role. These permissions/privileges are bound to the user only after the user has successfully authenticated. The TSF restricts the ability to manage the TSF data to only Security Administrators.

For the Command Center, the role of Security Administrator for the Web GUI is fulfilled by users assigned the "Admin" role and for in part by the "Analyst" role. When considering the scope of this evaluation, the only Security Administrator functionality of the Analyst role that pertains to the NDcPP is the analyst's ability to add and delete Sensor from the deployment of the TOE.

For the Command Center and Sensor, the role of Security Administrator for the CLI is fulfilled by the *silentdefense* role.

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

Management Function	Command Center Local CLI	Command Center Remote CLI (SSH)	Command Center Web GUI (HTTPS)	Sensor Local CLI	Sensor Remote CLI (SSH)
Configure Banner Text	silentdefense (vi issue)	silentdefense (vi issue.net)	Admin	Silentdefense (vi issue)	Silentdefense (vi issue.net)
Configure Idle Session Timeout	silentdefense (vi autologout.sh)	silentdefense (vi autologout.sh)	Admin	Silentdefense (vi autologout.sh)	Silentdefense (vi autologout.sh)
Initiate Manual Update	silentdefense <packagename> --verify-sig <digital signature value>	silentdefense <packagename> --verify-sig <digital signature value>	-	silentdefense <packagename> --verify-sig <digital signature value>	silentdefense <packagename> --verify-sig <digital signature value>
Configure Failed Lockout Threshold	silentdefense (vi faillock.conf)	silentdefense (vi faillock.conf)	Admin	Silentdefense (vi faillock.conf)	Silentdefense (vi faillock.conf)
Configure Lockout Duration	silentdefense (vi faillock.conf)	silentdefense (vi faillock.conf)	Admin	Silentdefense (vi faillock.conf)	Silentdefense (vi faillock.conf)
Manage the cryptographic keys	Refer to the last three rows of this table.	Refer to the last three rows of this table.	-.	Refer to the last three rows of this table.	Refer to the last three rows of this table.
Configure thresholds for SSH rekeying	silentdefense (vi sshd_config)	silentdefense (vi sshd_config)	-	Silentdefense (vi sshd_config)	Silentdefense (vi sshd_config)
Configure the interaction between TOE components (enablement)	-	-	Admin, Analyst	-	-
Configure System Time	silentdefense timedatectl set-time <Year-Month-Day> timedatectl set-time <Hour:Min:Second>	silentdefense timedatectl set-time <Year-Month-Day> timedatectl set-time <Hour:Min:Second>	Admin	Silentdefense timedatectl set-time <Year-Month-Day> timedatectl set-time <Hour:Min:Second>	Silentdefense timedatectl set-time <Year-Month-Day> timedatectl set-time <Hour:Min:Second>
Manage the TOE's trust store and designate X.509v3 certificates as trust anchors	-	-	-	silentdefense (/opt/nids-docker/states/nids-main/cert/<custom certs)	silentdefense (/opt/nids-docker/states/nids-main/cert/<custom certs)
Import X.509v3 certificates to the TOE's trust store	silentdefense (<Keystore path> /opt/sdconsole/ssl/ <Keystore filename> sd_keystore_web.pkcs12 <Tomcat config> /opt/sdconsole/tomcat/conf/server.xml)	silentdefense (<Keystore path> /opt/sdconsole/ssl/ <Keystore filename> sd_keystore_web.pkcs12 <Tomcat config> /opt/sdconsole/tomcat/conf/server.xml)	-	-	-

Manage trusted public keys database (user auth keys)	silentdefense (vi authorized_keys)	silentdefense (vi authorized_keys)	-	Silentdefense (vi authorized_keys)	Silentdefense (vi authorized_keys)

Table 23: Management Functions to Management Interface Identification

8.5.2 FMT_SMR.2

The TOE uses RBAC 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.

For the Command Center, the role of Security Administrator for the Web GUI is fulfilled by users assigned the "Admin" role and in part by the "Analyst" role. The Admin account by default has full view, edit, and read access to the TOE's resources. The Analyst role only maintains full view, edit, and read access for some TOE resources. When considering the scope of this evaluation, the only Security Administrator functionality of the Analyst role that pertains to the NDcPP is the analyst's ability to add and delete Sensor from the deployment of the TOE. A Security Administrator must assign permissions when creating any Web GUI users. To create an additional Web GUI Security Administrator, all the permissions must be selected and assigned to the user.

For the Command Center and Sensor, the role of Security Administrator for the CLI is fulfilled by the *silentdefense* role. The *silentdefense* user has full view, edit, read access to the configuration items, and full access to the underlying OS.

8.6 Protection of the TSF

8.6.1 FPT_APW_EXT.1

No user authentication passwords are stored by the TOE in plaintext. There is no function provided by the TOE to display an authentication password value in plaintext nor is the password data recoverable.

The SHA-256 hashed representation of the Web GUI user's password is stored on the Command Center.

The Command Center and Sensor have separately maintained credentials. Therefore, a SHA-512 hashed representation of the CLI user's password is stored on the respective TOE component for that user.

8.6.2 FPT_ITT.1

Communication between the Command Center and the Sensor is secured through the use of TLS channels. The Sensor is always the initiator of the connection. These TLS channels are utilized for the Command Center to manage the Sensors as well as for the Sensors to send network data and audit data to the Command Center.

8.6.3 FPT_SKP_EXT.1

Both the Command Center and the Sensor prevent the reading of all pre-shared keys, symmetric keys, and private keys. The TOE does not provide an interface for reading these keys. The X.509v3 certificate stored on the Command Center, the SSH host private key on both TOE components, the public keys used for SSH

communications on both TOE components, and the Command Center's public certificate used for TLS communication on the Sensor are all stored on their respective filesystems. These keys are protected by the TOE's role-based access control allowing them to be managed by only the Security Administrator, and direct access to private keys can only be read by the TOE itself. For SSH and TLS sessions, all pre-shared, symmetric keys and private keys are stored in volatile memory and are destroyed once the connection is closed.

8.6.4 FPT_STM_EXT.1

By default, the Command Center and the Sensor rely on the underlying Operating System's internal time settings. A Security Administrator has the ability to set the date and time manually through both the Command Center and the Sensor's respective local or remote CLIs using the timedatectl set-time command. Additionally, the Command Center's date and time settings can be configured manually through the Web GUI by a Security Administrator from within the Command Center's settings page. Date and time settings for the Sensors, within a deployment, can be altered through the Manage Sensors page in the Command Center Web GUI interface.

The TOE uses the clock for several security-relevant purposes, including:

- Audit record timestamps
- Frequency of sending audit data to the audit server
- Inactivity timeout for administrative sessions
- Expiration checking for certificates
- FIA_AFL.1 timer for lockout duration

8.6.5 FPT_TST_EXT.1

Upon startup, service restarts, and at the request of the authorized user, the TOE executes multiple self-tests to provide a mechanism of monitoring all TOE components and prevent whole component failure.

The following tests are part of the self-test suite:

#	Component	Validation	Fail Result
1.	Command Center & Sensor	Standard Linux Filesystem Check The TOE performs the following checks of the file system: • mounts (creates) basic virtual RAM file systems • verifies and mounts the non-volatile file system • verifies and mounts the active or standby software partition file system Failures for any of these checks could result in the platform entering a non-operational state or causing an automatic reboot to attempt to fix and continue startup.	Platform-fail
2.	Command Center & Sensor	Hardware Check The hardware check runs as part of system initialization which includes power-on self-tests of all the major hardware components (e.g., memory, CPU, Ethernet controllers) on the motherboard, including the components that connect to the buses. Failures for any of these checks will result in the platform entering a non-operational state or causing an automatic reboot to attempt to fix and continue startup.	Platform-fail
3.	Command Center & Sensor	Forescout GPG key is loaded & signature of POST 'truth' files are correct The 'truth' files include a file that contains SHA256 checksums of the TOE software modules and expected OS dependencies needed for the TOE to operate.	Software Hard-fail

#	Component	Validation	Fail Result
4.	Command Center & Sensor	Ubuntu LTS version is as expected Verifies that the correct version of Ubuntu is operating.	Software Hard-fail
5.	Command Center & Sensor	Validate integrity of cryptographic modules. The TOE validates the TOE software integrity by calculating the SHA256 checksum of the current TOE software modules (constant files such as executables, shared libraries, etc.). These results are compared to the known-good and signed values located in the 'truth' file verified in test 1.	Software Hard-fail
6.	Command Center & Sensor	OS-level dependencies of component are installed. The TOE verifies that the required dependencies and versions are installed on the OS against the list of expected dependencies defined in the signed file that is stored on the TOE.	Software Hard-fail

Table 24: Self-Test List with Failure Results

If a self-test failure were to occur that results in a Software Hard-fail, an audit record is generated and the TOE component will enter a maintenance state, meaning the TOE software is not operational. A failure that results in a Software Hard-fail result for a TOE component does not cause a complete shutdown of the system as a whole.

A CLI Security Administrator may execute the self-test checks (covered by 2 through 6 in Table 24) manually where the output is displayed on the screen.

These tests are sufficient to validate the correct operation of the TSF because they verify that the software has 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.

8.6.6 FPT_TUD_EXT.1

Forescout provides its customers access to new security updates via the Forescout portal. The updates are posted to the portal by Forescout as soon as they become available. Although notifications are pushed out to Security Administrators, it is recommended that the Forescout portal is checked periodically. Updates should be installed as soon as they become available.

Neither the Command Center nor Sensor automatically check for software or firmware updates. The Security Administrator is made aware of software updates via notices from the Forescout portal. The Command Center and the Sensor do not automatically download an update nor do they connect to the update server directly. The Security Administrator must initiate the software download from the Forescout portal onto the Remote Management Workstation. From there, each TOE components' software update is copied to the respective Command Center or Sensor. Updates to the Command Center and the Sensor can only be accomplished by a Security Administrator through each component's corresponding CLI; updates cannot be initiated through the Web GUI.

Update packages for the Command Center and the Sensor may include updates for either or both the TOE component's software and/or its underlying operating system. For Command Center and Sensor update packages, a checksum and digital signature check are performed on the software update in order to validate the authenticity of the update prior to installation. If the digital signature is not successfully validated, the update is not installed. Digital signature validation failures occur when the update file has been tampered with, an incorrect key was imported, an incorrect KEY_ID was passed, or the signature verification parameter is not included when executing the update. There is no administrative override mechanism

available for a Security Administrator to force the update to be installed on a TOE component if the digital signature is not validated.

When the Command Center successfully installs a software update, the Command Center reboots and all TSF performed by the Command Center are halted until the Command Center successfully reboots. In addition, any TSF performed by the Sensor which is reliant on an established channel with the Command Center is also halted until the Command Center successfully reboots and a secure channel is reestablished between the TOE components. When the Sensor successfully installs a software update, the Sensor reboots and all TSF performed by the Sensor are halted until the Sensor successfully reboots. In addition, any TSF performed by the Command Center which is reliant on an established channel with the Sensor is also halted until the Sensor successfully reboots and a secure channel is reestablished between the TOE components.

Upon login to the Web GUI, the licensing page displays the current software version of the Command Center. After a successful login to the Command Center, the current versions of the deployed Sensor can also be checked.

8.7 TOE Access

8.7.1 FTA_SSL_EXT.1

When a local session is inactive for the configured period of time, the TOE component terminates the session. The Security Administrator configures the inactivity timer by editing the *autologout.sh* file at the CLI. The inactivity timer must be configured separately for the Command Center and the Sensor using their corresponding CLI. The inactivity timeout period for local sessions has a default setting of 600 seconds.

8.7.2 FTA_SSL.3

For all components of the TOE, the TSF enforces termination of a remote user session after a defined period of inactivity. Reauthentication is required to establish a new remote session for both the Web GUI and CLI interfaces.

For both the Command Center and the Sensor components, the Security Administrator configures the remote CLI inactivity timer by editing the *faillock.conf* file at the CLI. The inactivity timer must be configured separately for the Command Center and the Sensor using the corresponding CLI. The inactivity timeout period for remote sessions has a default setting of 600 seconds.

For the Command Center's Web GUI interface, the automatic inactivity timeout threshold is configurable by a Security Administrator using the Web GUI and has a default of 10 minutes.

8.7.3 FTA_SSL.4

A Command Center Web GUI user can terminate their own user session by clicking the logout icon displayed in the top right corner of the application window. The display of the Web GUI login page is indicative of a successful logout. The user must reauthenticate in order to establish a new session within the Web GUI.

A Command Center or Sensor CLI user (local or remote) has the ability to terminate their own user session by typing "exit" in the command line and pressing enter. The display of a session closed message on the command line is indicative of a successful logout. The user must reauthenticate in order to establish a new local or remote CLI session.

8.7.4 FTA_TAB.1

The Command Center can be accessed locally using a physical connection, remotely using a SSH connection, or remotely using a HTTPS connection. The Sensors can be accessed locally using a physical connection or remotely using a SSH connection. All TOE components display a customizable pre-authentication warning banner for all user interfaces. The Security Administrator must customize the warning banner for each individual TOE component:

- The Web GUI warning banner is configured using the Command Center's Web GUI interface
- The Command Center local CLI warning banner is configured by editing the *issue* file on the Command Center
- The Command Center remote CLI warning banner is configured editing the *issue.net* file on the Command Center
- The Sensor local CLI warning banner is configured by editing the *issue* file on the Sensor
- The Sensor remote CLI warning banner is configured by editing the *issue.net* file on the Sensor

8.8 Trusted Path/Channels

8.8.1 FTP_ITC.1

The Command Center uses the SSH protocol to initiate and establish the trusted channel to export audit records to an audit server. The Command Center acts as a SSH client and is conformant to the requirements stated in FCS_SSHC_EXT.1.

8.8.2 FTP_TRP.1/Admin

Remote administration of the TOE is secured by the utilization of SSH and HTTPS protocols.

An HTTPS connection is used for establishing a connection from the Management Workstation to the Command Center's Web GUI for remote management. The Command Center acts as the HTTPS/TLS server and is conformant to the requirements stated in FCS_HTTPS_EXT.1 and FCS_TLSS_EXT.1.

An SSH connection is used for establishing a connection from the Management Workstation to the Command Center or the Sensor for remote management using the CLI. The TOE components act as a SSH server and are conformant to the requirements stated in FCS_SSHS_EXT.1.