Splunk Enterprise 7.3

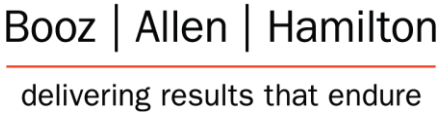
Security Target

ST Version: 1.2

January 23, 2020

Splunk
270 Brannan Street
San Francisco, CA 94107

Prepared By:



Cyber Assurance Testing Laboratory
1100 West Street
Laurel, MD 20707

Table of Contents

Table of Figures

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: Splunk Enterprise 7.3 Security Target
ST Version: 1.2
ST Publication Date: January 23, 2020
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 product-specific terminology used throughout this ST is defined in Table 1. Technology terms that are related to the security functionality claimed by the TOE are defined in the introductory materials of the claimed Protection Profile. These tables are to be used by the reader as a quick reference guide for terminology definitions.

Table 1: Customer Specific Terminology

Term	Definition
Security Administrator	A security administrator is an individual who has permissions to modify the behavior of the TOE. This includes the individual that installs it on the underlying platform but can also include other individuals if administrator access is granted to them on Splunk Web or Splunk CLI.
Splunk CLI	Splunk CLI is an application that can be used to interface with the TOE on the local system. It is launched from a shell.
Splunk Web	Splunk Web (or "Web UI") is a web-based application that can be used to manage the TOE remotely using HTTPS.
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 a security administrator uses to manage it (web browser, terminal client, etc.).
User	An individual who has access to the TOE but is not able to manage its behavior.

1.1.4 Acronyms

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

Table 2: Acronym Definition

Acronym	Definition
AES	Advanced Encryption Standard
ASLR	Address Space Layout Randomization
CA	Certification Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CLI	Command Line Interface
CRL	Certificate Revocation List
CSP	Critical Security Parameter
DHE	Diffie-Hellman Key Exchange
DRBG	Deterministic Random Bit Generator
ECDHE	Elliptic Curve Diffie-Hellman Key Exchange
ECDSA	Elliptic Curve Digital Signature Algorithm
GCM	Galois/Counter Mode
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol

IT	Information Technology
JIT	Just-in-Time (compilation)
OS	Operating System
OSP	Organizational Security Policy
PCRE	Perl Compatible Regular Expressions
PP	Protection Profile
NIAP	National Information Assurance Partnership
RA	Registration Authority
RBG	Random Bit Generator
RHEL	Red Hat Enterprise Linux
SAR	Security Assurance Requirement
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SMTP	Simple Mail Transfer Protocol
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
UI	User Interface

1.1.5 Reference

[1] Protection Profile for Application Software, version 1.2 (App PP)

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

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

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

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

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

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

[8] NIST Special Publication 800-56A Revision 2, Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography, August 2009

[9] NIST Special Publication 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007

[10] NIST Special Publication 800-57 Part 1 Revision 4 Recommendation for Key Management, January 2016

[11] FIPS PUB 186-4, Digital Signature Standard (DSS), July 2013

[12] FIPS PUB 180-4 Federal Information Processing Standards Publication Secure Hash Standard (SHS) March 2012

[13] NIST Special Publication 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators June 2015
[14] FIPS PUB 198-1 Federal Information Processing Standards Publication The Keyed-Hash Message Authentication Code (HMAC) July 2008
[15] Splunk Enterprise 7.3 Supplemental Administrative Guidance for Common Criteria Version 1.2

1.2 TOE Reference

The TOE is Splunk Enterprise 7.3, which is an application residing on a Linux OS.

1.3 TOE Overview

The Target of Evaluation (TOE) is the Splunk Enterprise 7.3 ("Splunk") application executing on a Linux operating system (OS). The primary function of Splunk is to collect system generated data from various types of platform systems and aggregate it in a centralized location for real-time visibility and analysis of system behavior. Additional operational functional behavior is dependent on whether the TOE has been configured to use the indexer or forwarder.

The indexer functionality is responsible for receiving data from trusted external sources such as databases, web services, and one or more additional instances of Splunk configured with the forwarder functionality enabled via HTTPS/TLS. Whereas, the forwarder functionality is responsible for transmitting the system-generated data to an external trusted entity such as an additional instance of Splunk configured with the indexer functionality enabled via HTTPS/TLS.

While the product vendor provides multiple versions of the product, only the full Linux version of Splunk Enterprise 7.3, operating on Red Hat Enterprise Linux (RHEL) and configured with either the indexer or forwarder functionality enabled, is considered the TOE – other product versions or platforms were not evaluated, and no security claims are made for them. In the evaluated configuration, Splunk Enterprise 7.3 is installed on top of the RHEL OS. When the TOE is configured with the indexer functionality (aka Splunk indexer), any Splunk forwarders are considered to be trusted non-TOE external transmitters (data feeds). When the TOE is configured with the forwarder functionality (aka Splunk forwarder), then the receiving Splunk indexer is considered to be a trusted non-TOE external data feed receiver.

The administrative interfaces include a local CLI and a web UI for remote access.

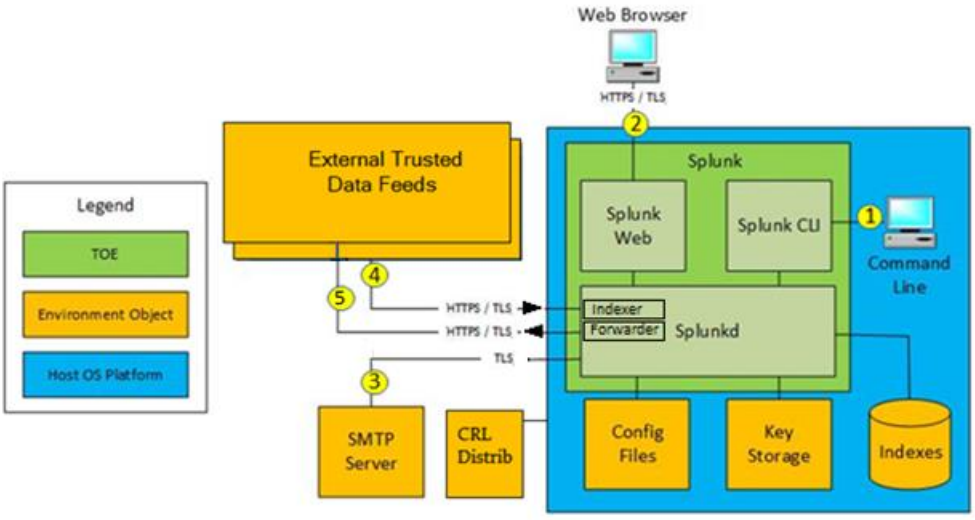


Figure 1: TOE Boundary

As illustrated in Figure 1, the TOE boundary contains 3 subsystems: Splunk Web, Splunk CLI, and Splunkd. Splunk Web and Splunk CLI are accessed through a supported web browser or command-line interface. Splunkd is the application process that provides most of the product functionality. The figure also depicts that the TOE uses the host operational environment for storing the TOE's configuration files, keystore, and datastore (index). The actual configuration files, keys, and data are considered part of the application.

Splunk CLI provides a command line interface to the product that can be accessed locally through a terminal application running on the host platform. It has the same functionality as the Splunk Web subsystem except for visual presentation functionality. A user uses this subsystem by navigating to the folder in which Splunkd resides. The user then issues the command "splunk" to run the executable along with any desired command-line arguments.

Splunk Web is a browser-based UI that supports the latest versions of Google Chrome, Mozilla Firefox, Apple Safari. Splunk Web also supports Internet Explorer 11 when not configured in Compatibility Mode. This transmission is secured using HTTPS/TLS. It provides a graphical interface to the product in order to perform remote administrative activities or to view reports or other graphical displays of system data that is collected by Splunkd. Users authenticate to Splunk Web using username and password. The Splunk Web component is only responsible for receiving user inputs; all authorizations are performed by Splunkd. The TOE is a TLS server for this interface.

Splunkd is the subsystem that consists of most of the functionality in the product. This subsystem handles identification, authentication, and authorization of users to access the application and interact with its administrative functions. The primary functionality of the product from a user perspective is to search accumulated data. A user issues a search command, which will search all of the indexes that the user has access to assuming they have the privilege to search. Every search entered by a user starts a new Splunkd process that only performs that search and returns the result to the parent Splunkd process. This data is then returned to the user, and there are a set of actions that a user can perform with this search and the search data. Because the focus of the claimed Protection Profile is the general security of the application and how it interfaces with its underlying platform, only the functionality that is part of the claimed

Protection Profile is considered to be part of the TOE. This functionality is described in more detail in section 2.5 below.

The TOE indexer will additionally interface with 2 external trusted IT entities:

- SMTP server: The TOE communicates with an SMTP server using TLS to send out configured alerts based on the TOE's analysis of the system generated data that it receives. The TOE is a TLS client for this interface (3).
- External trusted data feed: The TOE indexer communicates with an external trusted data feed using HTTPS/TLS to receive non-TSF related data to populate Splunk's local datastore (4). The external trusted data feed was a second Splunk Enterprise instantiation configured as a forwarder.

The TOE forwarder will additionally interface with 1 external trusted IT entities:

- External trusted data feed receiver: The TOE forwarder communicates with an external trusted data feed receiver to transmit non-TSF related data (5). The external trusted data feed receiver was a second Splunk Enterprise instantiation configured as an indexer.

NOTE: The external trusted data feed was a second Splunk Enterprise instantiation. When the TOE is configured as a Splunk indexer, any Splunk forwarders are considered to be part of the operational environment as trusted non-TOE external transmitters (data feeds). When the TOE is configured as a Splunk forwarder, then the receiving Splunk indexer is considered to be part of the operational environment as a trusted non-TOE external data feed receiver.

1.4 TOE Type

The TOE type for Splunk Enterprise 7.3, configured as either an indexer or forwarder, is Application Software. The *Protection Profile for Application Software* [App PP] specifies several use cases that conformant TOEs may implement. In particular the TOE supports:

Use Case 1, Content Creation is defined as follows: "The application allows a user to create content, saving it to either local or remote storage. Example content includes text documents, presentations, and images."

> Splunk indexer implicitly supports a user's ability to create content by creating/collecting system data from its host platform and storing it locally in a data store for the end user consumption.
>
> Splunk forwarder implicitly supports a user's ability to create content by creating/collecting system data from its host platform and storing it remotely, to such a device as a Splunk indexer, for the end user consumption.

Use Case 2, Content Consumption, is defined as follows: "The application allows a user to consume content, retrieving it from either local or remote storage."

> Splunk indexer is considered to implement content consumption because it allows a user to consume (query) log data stored on the local filesystem (Splunk indexer) and generate human-readable reports and views on this data.

2 TOE Description

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

2.1 Evaluated Components of the TOE

The TOE is the Splunk Enterprise 7.3 ("Splunk") application executing on a Linux OS. In the evaluated configuration, Splunk Enterprise 7.3 is installed on top of the RHEL OS and configured with either the indexer or forwarder functionality enabled. The administrative interfaces include a local CLI and a web UI for remote access.

The TOE indexer was configured to securely communicate with the following external IT entities: SMTP server (TOE acts as client only), external a trusted data feed (TOE acts as server), and a management workstation (TOE acts as server). The external trusted data feed was an instantiation of Splunk software configured as a forwarder and is considered part of the operational environment for the TOE indexer.

The TOE forwarder was configured to securely communicate with the following external IT entities: external a trusted data receiver (TOE acts as client), and a management workstation (server). The external trusted data feed receiver was an instantiation of Splunk software configured as an indexer and is considered part of the operating environment for the TOE forwarder.

All claimed PP related functionality is contained whether Splunk is configured as an indexer or a forwarder.

2.2 Components and Applications in the Operational Environment

The following table lists components and applications in the TOE's operational environment that must be present for the TOE to be operating in its evaluated configuration:

Table 3: Components of the Operational Environment

Component	Definition
External Trusted Data Feed	External data source for transmitting non-TSF related data to the TOE indexer for populating Splunk's datastore. The external data source must use HTTPS/TLS to communicate with the TOE.
External Trusted Data Feed Receiver	External data source for receiving non-TSF related data from the TOE forwarder. The external data source must use HTTPS/TLS to communicate with the TOE.
Host Platform	A general-purpose computer on which the Linux operating system and the TOE is installed. The TOE requires network resources from the host platform.
Management Workstation	Any general-purpose computer that is used by a security administrator to manage the TOE remotely via a web browser. Note that the host platform can also be used to administer the TOE locally.
SMTP Server	An email server that can receive alerts from the TOE and deliver them to users in the Operational Environment via email.
CRL Distribution Point	A server that provides updated revocation lists for the TOE's certificate validation functionality.

2.3 Excluded from the TOE

The following optional products, components, and/or applications can be integrated with the TOE but 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 optional components that are omitted from the installation process.

2.3.2 Installed but Requires a Separate License

The following components are included with the Splunk Enterprise 7.3 product but are separately licensed and not considered to be within the TOE boundary:

- Hunk: Splunk analytics support for Hadoop – in the TOE's evaluated configuration this license will not be applied, and this component will not be active.

2.3.3 Installed But Not Part of the TSF

This section contains functionality or components that are part of the purchased product but are not part of the TSF relevant functionality that is being evaluated as the TOE.

- HTTP administrative interface – in the evaluated configuration, the TOE will be configured to only permit HTTPS remote communications.
- The TOE's ability to search and index information is not part of the evaluation. However, the data is needed in order to stimulate events for testing PP related functionality.

Additionally, the TOE includes several other functions that are outside the scope of the claimed Protection Profile. These functions, such as non-TSF data importing/exporting, have no SFRs that apply to them and are not included in the scope of the evaluation.

2.4 Physical Boundary

2.4.1 Hardware

Splunk Enterprise 7.3 is a software-only TOE. All hardware that is present is part of the TOE's Operational Environment. The following system configuration was used for the testing of the TOE (both indexer and forwarder):

- Red Hat Enterprise Linux 6.5 64 bit
- Intel(R) Xeon(R) CPU E3-1220 v3 @ 3.10GHz
- 16 GB RAM
- 500 GB disk

2.4.2 Software

The physical boundary of the TOE software is the Splunk application including configuration files, keys, and data. This also includes the Splunkd application process, Splunk Web, and the Splunk CLI administrative interfaces as depicted in Figure 1 above. The TOE software includes OpenSSL which performs the TOE's cryptographic operations.

2.5 Logical Boundary

The TOE is comprised of the following security features as scoped by the App PP:

- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Privacy
- Protection of the TSF
- Trusted Path/Channel

2.5.1 Cryptographic Support

The TOE software includes OpenSSL which performs the TOE's cryptographic operations required to support the establishment of trusted channels and paths to protect data in transit. As an application on an operating system, the TOE interfaces with the operating system's key storage to securely store key data related to secure communications. The TOE also relies on the underlying platform to generate entropy that is used as input data for the TOE's deterministic random bit generator (DRBG).

The following table contains the CAVP algorithm certificates:

Table 4: Cryptographic Algorithm Table

SFR	Cert Name (Claimed Algorithm)	CAVP Cert. #
FCS_CKM_EXT.1 Key generation	ECDSA: 186-4 Key Pair Generation and Private Key Validation (P-256, P-384, P-521)	C948
FCS_CKM.1 Asymmetric key generation	ECDSA: 186-4 Key Pair Generation and Private Key Validation (P-256, P-384, P-521)	C948
FCS_CKM.2 Key establishment	ECDHE: KAS-ECC (P-256, P-384, P-521)	C948
FCS_COP.1(1) Encryption/decryption	AES (CBC-256, GCM-128 GCM-256)	C948
FCS_COP.1(2) Hash	SHS (SHA-256, SHA 384, SHA-512)	C948
FCS_COP.1(3) Signing and verification	ECDSA: Signature Generation and Signature Verification (P-256: SHA-256, SHA-384, SHA512 P-384: SHA-256, SHA-384, SHA512 P-521: SHA-256, SHA-384, SHA512)	C948
FCS_COP.1(4) Keyed-hash message authentication	HMAC (HMAC-SHA-256, HMAC-SHA-384)	C948
FCS_RBG_EXT.1 Random Bit Generation	DRBG (CTR-DRBG)	C948
	AES (CTR-AES-256)	C948

2.5.2 User Data Protection

In the evaluated configuration, the TOE will reside on an encrypted disk partition on the underlying platform to secure its data at rest. The TOE protects data stored on the underlying platform by minimizing its use of platform resources. Specifically, the TOE only requires the use of the underlying platform's network connectivity for administrative activities, email alerts, receipt and transmission of non-TSF related data from/to external trusted data feeds.

2.5.3 Identification and Authentication

In order to facilitate secure communications using HTTPS/TLS, the TOE provides a mechanism to validate X.509 certificates. While the HTTPS/TLS implementation will automatically reject a certificate if it is found to be invalid, a certificate with unknown revocation status (because the TSF is unable to read the CRL) is accepted.

2.5.4 Security Management

The TOE does not provide any default credential used for initial authentication. The files and directories that comprise the TOE are protected against unauthorized access by only permitting write access to the user that performed the installation. The TOE uses the underlying platform's recommended methods for storing and setting configuration options. The TOE also provides the security administrators with the ability to configure the supported TLS cipher suites of the trusted channels and query the existing TOE software version.

2.5.5 Privacy

The TOE ensures the privacy of its security administrators and users by not providing any capability to transmit personally identifiable information (PII) over the network.

2.5.6 Protection of the TSF

The TOE protects against exploitation by implementing address space layout randomization (ASLR) and only allocating memory for both writing and execution for just-in-time (JIT) compilation. The TOE is also compatible with SELinux and is compiled with stack-based buffer overflow protection. It also prevents the writing of user-modifiable files to directories that contain executable files.

The TOE uses standard platform APIs and includes only the third-party libraries it needs to perform its functionality. The TOE version can be checked either through its management interfaces or through the underlying platform's package manager. Updates must be manually downloaded to the platform's file system and installed using the platform's package manager. In the evaluated configuration, the security administrator will download and install a public key from the TOE's developer that is installed into the package manager and used to verify the integrity of any updates to the TOE.

2.5.7 Trusted Path/Channel

The TOE protects all data in transit using HTTPS over TLS or standalone TLS. HTTPS/TLS protocol is used to secure remote administration using the web UI. The TOE, acting as an indexer, uses TLS to securely send alerts to a remote SMTP server in the Operational Environment. HTTPS/TLS is used to secure communications between the TOE indexer and external trusted data feeds. Additionally, the TOE

forwarder requires the use of HTTPS/TLS to secure communications for transmitting data to an external trusts data feed receiver.

3 Conformance Claims

3.1 CC Version

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

3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 2 extended to include all applicable NIAP and International interpretations through January 23, 2020.

3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 3 extended to include all applicable NIAP and International interpretations through January 23, 2020.

3.4 PP Claims

This ST claims exact conformance to the following Protection Profile:

- Protection Profile for Application Software, version 1.2 [App PP]

3.5 Package Claims

The TOE claims exact conformance to the App PP, version 1.2, which is conformant with CC Part 3.

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

- FCS_CKM_EXT.1
- FCS_CMK.1(1)
- FCS_CKM.2
- FCS_COP.1(1)
- FCS_COP.1(2)
- FCS_COP.1(3)
- FCS_COP.1(4)
- FCS_HTTPS_EXT.1
- FCS_RBG_EXT.2
- FCS_TLSC_EXT.1
- FCS_TLSC_EXT.4
- FCS_TLSS_EXT.1
- FIA_X509_EXT.1
- FIA_X509_EXT.2

The TOE claims FCS_TLSC_EXT.2 optional SFR.

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

3.6 Package Name Conformant or Package Name Augmented

This ST and TOE are in exact conformance with the App PP.

3.7 Conformance Claim Rationale

The App PP states the following: “The requirements in this document apply to application software which runs on mobile devices ("apps"), as well as on desktop and server platforms. Some application types are covered by more specific PPs, which may be expressed as Extended Packages of this PP. Such applications are subject to the requirements of both this PP and the Extended Package that addresses their special functionality. PPs for some particularly specialized applications may not be expressed as EPs at this time, though the requirements in this document should be seen as objectives for those highly specialized applications.

Although the requirements in this document apply to a wide range of application software, consult guidance from the relevant national schemes to determine when formal Common Criteria evaluation is expected for a particular type of application. This may vary depending upon the nature of the security functionality of the application.”

The TOE is a standalone application which runs on a desktop/server Linux platform and is therefore considered to be relevant to the App PP. There are no Extended Packages to the App PP that are applicable to Splunk, so the TOE is characterized only as a software application.

3.8 Technical Decisions

The following table is a complete list of Technical Decisions that apply to the App PP v1.2 and the evaluation activities that must be performed during the evaluation of this TOE. Note that Technical Decisions were not considered to be applicable if any of the following conditions were true:

- The Technical Decision does not apply to the App PP.
- The Technical Decision does not apply to the current version of the App PP.
- The Technical Decision applies to an SFR that was not claimed by the TOE.
- The Technical Decision applies to an SFR selection or assignment that was not chosen for the TOE.
- The Technical Decision only applies to one or more Application Notes in the App PP and does not affect the SFRs or how the evaluation of the TOE is conducted.
- The Technical Decision is an affirmation that an existing requirement or Evaluation Activity is correct.
- The Technical Decision was superseded by a more recent Technical Decision.
- The Technical Decision is issued as guidance for future versions of the App PP.

Table 5: Technical Decisions

TD #	Title	References	Changes			Analysis to this evaluation	
			SFR	AA	Notes	NA	Reason
TD0435	Alternative to SELinux for FPT_AEX_EXT.1.3	FPT_AEX_EXT.1.3		X			AA: Test wording for Linux

TD0434	Windows Desktop Applications Test	FDP_DEC_EXT.1.1		X		X	AA: Test wording Not claiming Windows
TD0427	Reliable Time Source	A.Platform	X				Updated wording to Assumption.
TD0392	FCS_TLSC_EXT.1.2 Wildcard Checking	FCS_TLSC_EXT.1.2		X			AA: Test wording
TD0390	Cryptographically Secure RNG	FCS_RBG_EXT.1		X		X	Supersedes TD0172 Not claiming Windows
TD0389	Handling of SSH EP claim for platform	FTP_DIT_EXT.1.1	X	X	X	X	Supersedes TD0177 Not claiming SSH Wording of ST was verified.
TD0385	FTP_DIT_EXT.1 Assurance Activity Clarification	FTP_DIT_EXT.1		X		X	Not claiming VPN or IPSEC.
TD0382	Configuration Storage Options for Apps	FMT_MEC_EXT.1		X			AA_ Test wording Claiming Linux
TD0380	Linux Keyring Requirement in FCS_STO_EXT.1	FCS_STO_EXT.1			X		Supersedes TD0192 Claiming Linux
TD0364	Android mmap testing for FPT_AEX_EXT.1.1	FCS_AEX_EXT.1.1		X		X	AA: Test wording Not claiming Android
TD0359	Buffer Protection	FPT_AEX_EXT.1.5		X		X	Not claiming Windows
TD0358	Cipher Suites for TLS in SWApp v1.2	FCS_TLSC_EXT.1.1 and FCS_TLSS_EXT.1.1	X		X		Supersedes TD0283
TD0327	Default file permissions for FMT_CFG_EXT.1.2	FMT_CFG_EXT.1.2	X	X	X		AA: Test wording ST SFR updated
TD0326	RSA-based key establishment schemes	FCS_CKM.2.1 and FCS_TLSS_EXT.1.3	X		X		Supersedes TD0293 and TD0107
TD0305	Handling of TLS connections with and without mutual authentication	FCS_TLSC_EXT.2.1		X			AA: Test Wording
TD0304	Update to FCS_TLSC_EXT.1.2	FCS_TLSC_EXT.1.2		X			AA: Test Wording
TD0300	Sensitive Data in FDP_DAR_EXT.1	FDP_DAR_EXT.1	X	X	X		AA: TSS wording
TD0296	Update to FCS_HTTPS_EXT.1.3	FCS_HTTPS_EXT.1.3	X	X	X		AA: Test wording
TD0295	Update to FPT_AEX_EXT.1.3 Assurance Activities	FPT_AEX_EXT.1.3		X		X	Supersedes TD0269 AA: Test wording Not claiming Android or

							Windows.
TD0268	FMT_MEC_EXT.1 Clarification	FMT_MEC_EXT.1			X		
TD0267	TLSS testing - Empty Certificate Authorities list	FCS_TLSS_EXT.1.5		X			AA: Test wording
TD0244	FCS_TLSC_EXT - TLS Client Curves Allowed	FCS_TLSC_EXT.4.1	X	X	X		AA: Test wording
TD0241	Removal of Test 4.1 in FCS_TLSS_EXT.1.1	FCS_TLSS_EXT.1.1		X			AA: Test wording
TD0238	User-modifiable files FPT_AEX_EXT.1.4	FPT_AEX_EXT.1.4		X			AA: Test wording
TD0221	FMT_SMF.1.1 - Assignments moved to Selections	FMT_SMF.1.1	X			X	Not claiming SWFE EP conformance Supersedes TD0122
TD0217	Compliance to RFC5759 and RFC5280 for using CRLs	FIA_X509_EXT.1.1	X				Selecting 5759
TD0215	Update to FCS_HTTPS_EXT.1.2	FCS_HTTPS_EXT.1.2	X	X			AA: Test wording
TD0178	Integrity for installation tests in AppSW PP	FPT_TUD_EXT.1.3		X			AA: Test wording
TD0177	FCS_TLSS_EXT.1 Application Note Update	FTP_DIT_EXT.1			X		
TD0174	Optional Ciphersuites for TLS	FCS_TLSC_EXT.1.1	X				
TD0163	Update to FCS_TLSC_EXT.1.1 Test 5.4 and FCS_TLSS_EXT.1.1 Test	FCS_TLSC_EXT.1.1, FCS_TLSS_EXT.1.1		X			AA: Test wording
TD0131	Update to FCS_TLSS_EXT.1.1 Test 4.5	FCS_TLSS_EXT.1.1		X			AA: Test wording
TD0121	FMT_MEC_EXT.1.1 Configuration Options	FMT_MEC_EXT.1	X	X	X	X	Not claiming SWFE EP conformance
TD0119	FCS_STO_EXT.1.1 in PP_APP_v1.2	FCS_STO_EXT.1.1	X	X	X		AA: TSS and Testing

4 Security Problem Definition

4.1 Threats

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

Table 6: TOE Threats

Threat	Threat Definition
T.NETWORK_ATTACK	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.
T.NETWORK_EAVESDROP	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.
T.LOCAL_ATTACK	An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.
T.PHYSICAL_ACCESS	An attacker may try to access sensitive data at rest.

4.2 Organizational Security Policies

There are no Organizational Security Policies in the App PP.

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 App PP.

Table 7: TOE Assumptions

Assumption	Assumption Definition
A.PLATFORM[1]	The TOE relies upon a trustworthy computing platform with a reliable time clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.
A.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.
A.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

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.

[1] TD0427

4.4.1 TOE Security Objectives

This section identifies the security objectives of the TOE as defined by the App PP.

Table 8: TOE Objectives

Objective	Objective Definition
O.INTEGRITY	Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom if ever shipped without errors, and the ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options.
O.QUALITY	To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs.
O.MANAGEMENT	To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII.
O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.
O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application.

4.4.2 Security Objectives for the Operational Environment

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

Table 9: TOE Operational Environment Objectives

Objective	Objective Definition
OE.PLATFORM	The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.
OE.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.
OE.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

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.

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 and italicized text.
- **Selection:** allows the specification of one or more elements from a list. Indicated with underlined text.
- **Iteration:** allows a component to be used more than once with varying operations. Indicated with a sequential number in parentheses following the element number of the iterated SFR.

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:

Table 10: Security Functional Requirements for the TOE

Class Name	Component Identification	Component Name
Cryptographic Support	FCS_CKM_EXT.1	Cryptographic Key Generation Services
	FCS_CKM.1(1)	Cryptographic Asymmetric Key Generation
	FCS_CKM.2	Cryptographic Key Establishment
	FCS_COP.1(1)	Cryptographic Operation – Encryption/Decryption
	FCS_COP.1(2)	Cryptographic Operation – Hashing
	FCS_COP.1(3)	Cryptographic Operation – Signing
	FCS_COP.1(4)	Cryptographic Operation – Keyed-Hash Message Authentication
	FCS_HTTPS_EXT.1	HTTPS Protocol
	FCS_RBG_EXT.1	Random Bit Generation Services
	FCS_RBG_EXT.2	Random Bit Generation from Application
	FCS_STO_EXT.1	Storage of Credentials
	FCS_TLSC_EXT.1	TLS Client Protocol
	FCS_TLSC_EXT.2	TLS Client Protocol
	FCS_TLSC_EXT.4	TLS Client Protocol
	FCS_TLSS_EXT.1	TLS Server Protocol
User Data Protection	FDP_DAR_EXT.1	Encryption of Sensitive Application Data
	FDP_DEC_EXT.1	Access to Platform Resources

	FDP_NET_EXT.1	Network Communications
Identification and Authentication	FIA_X509_EXT.1	X.509 Certificate Validation
	FIA_X509_EXT.2	X.509 Certificate Authentication
Security Management	FMT_CFG_EXT.1	Secure by Default Configuration
	FMT_MEC_EXT.1	Supported Configuration Mechanism
	FMT_SMF.1	Specification of Management Functions
Privacy	FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information
Protection of the TSF	FPT_AEX_EXT.1	Anti-Exploitation Capabilities
	FPT_API_EXT.1	Use of Supported Services and APIs
	FPT_LIB_EXT.1	Use of Third Party Libraries
	FPT_TUD_EXT.1	Integrity for Installation and Update
Trusted Path/Channel	FTP_DIT_EXT.1	Protection of Data in Transit

6.3 Security Functional Requirements

6.3.1 Class FCS: Cryptographic Support

6.3.1.1 *FCS_CKM_EXT.1 Cryptographic Key Generation Services*

FCS_CKM_EXT.1.1

The application shall [implement asymmetric key generation].

6.3.1.2 *FCS_CKM.1(1) Cryptographic Asymmetric Key Generation*

FCS_CKM.1.1(1)[2]

The application shall [implement functionality] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [

ECC schemes using ["NIST curves" P-256, P-384 and [P-521]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4] ,

] .

6.3.1.3 *FCS_CKM.2 Cryptographic Key Establishment*

FCS_CKM.2.1[3]

The application shall [implement functionality] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"

].

6.3.1.4 *FCS_COP.1(1) Cryptographic Operation – Encryption/Decryption*

FCS_COP.1.1(1)

The application shall perform encryption/decryption in accordance with a specified cryptographic algorithm

- AES-CBC (as defined in NIST SP 800-38A) mode;

and [AES-GCM (as defined in NIST SP 800-38D)]

and cryptographic key sizes 256-bit and [128-bit].

[2] TD0326
[3] TD0326

6.3.1.5 *FCS_COP.1(2) Cryptographic Operation – Hashing*

FCS_COP.1.1(2)

The application 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: FIPS Pub 180-4.

6.3.1.6 *FCS_COP.1(3) Cryptographic Operation – Signing*

FCS_COP.1.1(3)

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

[ECDSA schemes using "NIST curves" P-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5].

6.3.1.7 *FCS_COP.1(4) Cryptographic Operation – Keyed-Hash Message Authentication*

FCS_COP.1.1(4)

The application shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm

- HMAC-SHA-256

and [SHA-384] with key sizes [*equal to block sizes*] and message digest sizes 256 and [384] bits that meet the following: FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard.

6.3.1.8 *FCS_HTTPS_EXT.1 HTTPS Protocol*

FCS_HTTPS_EXT.1.1

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

FCS_HTTPS_EXT.1.2[4]

The application shall implement HTTPS using TLS in accordance with [FCS_TLSC_EXT.1, FCS_TLSS_EXT.1].

FCS_HTTPS_EXT.1.3[5]

The application shall [not establish the connection] if the peer certificate is deemed invalid.

6.3.1.9 *FCS_RBG_EXT.1 Random Bit Generation Services*

FCS_RBG_EXT.1.1

[4] TD0215
[5] TD0296

The application shall [implement DRBG functionality] for its cryptographic operations.

6.3.1.10 *FCS_RBG_EXT.2 Random Bit Generation from Application*

FCS_RBG_EXT.2.1

The application shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [CTR_DRBG (AES)].

FCS_RBG_EXT.2.2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [no other noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

6.3.1.11 *FCS_STO_EXT.1 Storage of Credentials*

FCS_STO_EXT.1.1[6]

The application shall [invoke the functionality provided by the platform to securely store [*credentials defined in Table 11*]] to nonvolatile memory.

6.3.1.12 *FCS_TLSC_EXT.1 TLS Client Protocol*

FCS_TLSC_EXT.1.1[7]

The application shall [implement TLS 1.2 (RFC 5246)] supporting the following cipher suites: [

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

FCS_TLSC_EXT.1.2

The application shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3

The application shall establish a trusted channel only if the peer certificate is valid.

6.3.1.13 *FCS_TLSC_EXT.2 TLS Client Protocol*

FCS_TLSC_EXT.2.1

The application shall support mutual authentication using X.509v3 certificates.

[6] TD0119
[7] TD0358 and TD0174

6.3.1.14 *FCS_TLSC_EXT.4 TLS Client Protocol*

FCS_TLSC_EXT.4.1[8]

The application shall present the supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [secp256r1, secp384r1, secp521r1].

6.3.1.15 *FCS_TLSS_EXT.1 TLS Server Protocol*

FCS_TLSS_EXT.1.1[9]

The application shall [implement TLS 1.2 (RFC 5246)] supporting the following cipher suites: [

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289]

and no other cipher suite.

FCS_TLSS_EXT.1.2

The application shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1 and [none].

FCS_TLSS_EXT.1.3[10]

The application shall generate key establishment parameters using [ECDHE over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves].

FCS_TLSS_EXT.1.4

The application shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.1.5

The application shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSS_EXT.1.6

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

6.3.2 Class FDP: User Data Protection

6.3.2.1 *FDP_DAR_EXT.1 Encryption of Sensitive Application Data*

FDP_DAR_EXT.1.1[11]

The application shall [leverage platform-provided functionality to encrypt sensitive data] in non-

[8] TD0244
[9] TD0358
[10] TD0326
[11] TD0300

volatile memory.

6.3.2.2 *FDP_DEC_EXT.1 Access to Platform Resources*

FDP_DEC_EXT.1.1

The application shall restrict its access to [network connectivity].

FDP_DEC_EXT.1.2

The application shall restrict its access to [no sensitive information repositories].

6.3.2.3 *FDP_NET_EXT.1 Network Communications*

FDP_NET_EXT.1.1

The application shall restrict network communication to [respond to [*remote administration requests (web server), receipt of non-TSF related data from/to external trusted data feeds (indexer functionality)*], [*transmission of alerts to environmental SMTP server, transmission of non-TSF related data from/to external trusted data feeds (forwarder functionality*]].

6.3.3 Class FIA: Identification and Authentication

6.3.3.1 *FIA_X509_EXT.1 X.509 Certificate Validation*

FIA_X509_EXT.1.1[12]

The application shall [implement functionality] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate 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 application shall validate the revocation status of the certificate using [a Certificate Revocation List (CRL) as specified in RFC 5759].
- The application shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 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.
    - S/MIME certificates presented for email encryption and signature shall have

[12] TD0217

the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) 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.
- Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field.

FIA_X509_EXT.1.2

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

6.3.3.2 *FIA_X509_EXT.2 X.509 Certificate Authentication*

FIA_X509_EXT.2.1

The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [HTTPS, TLS].

FIA_X509_EXT.2.2

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

6.3.4 Class FMT: Security Management

6.3.4.1 *FMT_CFG_EXT.1 Secure by Default Configuration*

FMT_CFG_EXT.1.1

The application shall only provide enough functionality to set new credentials when configured with default credentials or no credentials.

FMT_CFG_EXT.1.2[13]

The application shall be configured by default with file permissions which protect the application's binaries and data files from modification by normal unprivileged user.

6.3.4.2 *FMT_MEC_EXT.1 Supported Configuration Mechanism*

FMT_MEC_EXT.1.1

The application shall invoke the mechanisms recommended by the platform vendor for storing and setting configuration options.

[13] TD0327

6.3.4.3 *FMT_SMF.1 Specification of Management Functions*

FMT_SMF.1.1

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

[[*enable/disable supported TLS cipher suites, and query the version of the TOE*]].

6.3.5 Class FPR: Privacy

6.3.5.1 *FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information*

FPR_ANO_EXT.1.1

The application shall [not transmit PII over a network].

6.3.6 Class FPT: Protection of the TSF

6.3.6.1 *FPT_AEX_EXT.1 Anti-Exploitation Capabilities*

FPT_AEX_EXT.1.1

The application shall not request to map memory at an explicit address except for [*none*].

FPT_AEX_EXT.1.2

The application shall [allocate memory regions with write and execute permissions for only [*just-in-time compilation functions sljit, libffi, luajit*]].

FPT_AEX_EXT.1.3

The application shall be compatible with security features provided by the platform vendor.

FPT_AEX_EXT.1.4

The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

FPT_AEX_EXT.1.5

The application shall be compiled with stack-based buffer overflow protection enabled.

6.3.6.2 *FPT_API_EXT.1 Use of Supported Services and APIs*

FPT_API_EXT.1.1

The application shall use only documented platform APIs.

6.3.6.3 *FPT_LIB_EXT.1 Use of Third Party Libraries*

FPT_LIB_EXT.1.1

The application shall be packaged with only [*third-party libraries listed in Table 14*].

6.3.6.4 *FPT_TUD_EXT.1 Integrity for Installation and Update*

FPT_TUD_EXT.1.1

The application shall [provide the ability] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2

The application shall be distributed using the format of the platform-supported package manager.

FPT_TUD_EXT.1.3

The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.

FPT_TUD_EXT.1.4

The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.5

The application shall [provide the ability, leverage the platform] to query the current version of the application software.

FPT_TUD_EXT.1.6

The application installation package and its updates shall be digitally signed such that its platform can cryptographically verify them prior to installation.

6.3.7 Class FTP: Trusted Path/Channel

6.3.7.1 *FTP_DIT_EXT.1 Protection of Data in Transit*

FTP_DIT_EXT.1.1[14]

The application shall [encrypt all transmitted data with [HTTPS, TLS]] between itself and another trusted IT product.

6.4 Statement of Security Functional Requirements Consistency

The Security Functional Requirements included in the ST represent all required SFRs specified in the claimed PP as well as a subset of the 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.

[14] TD0389

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 App PP.

7.1 Class ASE: Security Target

As per ASE activities defined in [CEM]

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.

7.4.3 Timely Security Updates (ALC_TSU_EXT.1)

7.4.3.1 *Developer Actions Element:*

ALC_TSU_EXT.1.1D

The developer shall provide a description in the TSS of how timely security updates are made to the TOE.

ALC_TSU_EXT.1.2D

The developer shall provide a description in the TSS of how users are notified when updates change security properties or the configuration of the product.

7.4.3.2 *Content and presentation elements:*

ALC_TSU_EXT.1.1C

The description shall include the process for creating and deploying security updates for the TOE software.

ALC_TSU_EXT.1.1C

The description shall express the time window as the length of time, in days, between public disclosure of a vulnerability and the public availability of security updates to the TOE.

ALC_TSU_EXT.1.1C

The description shall include the mechanisms publicly available for reporting security issues pertaining to the TOE.

7.4.3.3 *Evaluator action elements:*

ALC_TSU_EXT.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

8.1 Cryptographic Support

8.1.1 FCS_CKM_EXT.1 and FCS_CKM.1(1):

The TOE uses asymmetric cryptography in support of HTTPS/TLS trusted communications. However, all the asymmetric keys (certificates) used are created in the Operational Environment on a separate machine than the TOE. These certificates must be installed on the TOE during the installation process.

For TLS communications, the TOE implements the additional key generation functionality in support of ECC schemes using NIST curves P-256, P-384, and P-521 that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4. This function is handled by the TOE's OpenSSL cryptographic module and has been certified by CAVP under ECDSA PKV certificate #C948.

8.1.2 FCS_CKM.2:

The TOE supports Elliptic curve-based key establishment schemes for establishment of HTTPS/TLS communications. Elliptic curve-based key establishment conforms to NIST SP 800-56A. This function is handled by the TOE's OpenSSL cryptographic module and has been certified by CAVP under ECDHE:KAS certificate #C948

8.1.3 FCS_COP.1(1):

The TOE performs AES encryption in support of HTTPS/TLS communications. AES-CBC key size 256 and AES-GCM 128-bit and 256-bit keys are supported. This functionality is handled by the TOE's OpenSSL cryptographic module and has been certified by CAVP under AES certificate #C948.

8.1.4 FCS_COP.1(2):

The TOE performs cryptographic hashing in support of HTTPS/TLS. The SHA-256, SHA-384, and SHA-512 algorithms are supported. This functionality is handled by the TOE's OpenSSL cryptographic module and has been certified by CAVP under SHS certificate #C948.

8.1.5 FCS_COP.1(3):

The TOE provides digital signature services in support of validation of software updates and X.509v3 authentication. The TOE supports ECDSA schemes using NIST curves P-256, P-384, and P-521. This functionality is handled by the TOE's OpenSSL cryptographic module and has been certified by CAVP under ECDSA certificate #C948.

8.1.6 FCS_COP.1(4):

The TOE provides keyed-hash message authentication services in support of HTTPS/TLS communications. The TOE supports both HMAC-SHA-256 and HMAC-SHA-384 with key size equal to block size. This functionality is handled by the TOE's OpenSSL cryptographic module and has been certified by CAVP under HMAC certificate #C948.

8.1.7 FCS_HTTPS_EXT.1:

The TOE implements HTTPS/TLS in order to support secure communications for the following external interfaces:

- Remote security administrator to TOE (via web UI)
- TOE indexer from trusted data feeds (receipt)
- TOE forwarder to trusted data feed receiver (transmission)

Both interfaces use the same HTTPS implementation, which relies on the TLS client and server capability that is described in section 8.1.10. The TOE is automatically set to reject a connection in the event that a peer certificate is found to be invalid and this behavior is not configurable. Note that this is different from the behavior of the TSF if it is unable to determine the revocation status of a certificate, which is described in section 8.3.2.

8.1.8 FCS_RBG_EXT.1 and FCS_RBG_EXT.2:

The TOE implements its own DRBG functionality but collects entropy from the underlying platform (RDRAND on-chip circuit) as a seed. The TOE includes an OpenSSL implementation of the AES_CTR DRBG (CAVP DRBG certificate #C948 and AES certificate #C948) which is invoked by the TOE for random bit generation services by default. There is no ability to specify the use of an alternative DRBG. The Intel Xeon x64 processor provides access to the Intel RDRAND instruction which produces 64 bits of entropy output each time the RDRAND instruction is called. Whenever entropy is requested, RDRAND is called a sufficient number of times to provide the requested entropy on an as needed basis. This entropy is fed directly into the CTR_DRBG provided by OpenSSL. The amount of entropy that is collected is based on the function that the DRBG is being used for. In all cases, this amount is greater than or equal to the security strength of the data that is being output (e.g. a 256-bit AES key generation operation will collect at least 256 bits of entropy before the DRBG is invoked). The entropy source is described in greater detail in the proprietary Entropy Assessment Report.

8.1.9 FCS_STO_EXT.1:

The TOE stores all credential data in the GNOME keyring on the underlying platform. This includes password data as well as passphrases that protect private keys. In order to unlock the keyring, the security administrator is prompted for a passphrase during initial startup of the TOE. To write data to the GNOME keyring, the TSF provides the ‘splunk secret-storage’ command at the CLI.

Private keys are stored in encrypted format on the file system of the underlying platform. The credentials to unlock these keys are stored in the keyring.

The following table includes the credentials that are stored in the keyring and their purpose:

Table 11: Credentials Stored in Keyring

Configuration Item	Stanza	Parameter	Description
alert_actions	email	auth_password	SMTP password for sending email alerts to an external SMTP server. This was only used when Splunk was configured as an indexer in the evaluated configuration.

server	sslConfig	sslPassword	Passphrase protecting splunkd server private key. Required.
server	kvstore	sslPassword	Passphrase protecting key-value store private key. Required.
web	settings	sslPassword	Passphrase protecting the web server's private key. This is only used if Splunk is being configured to use the web UI using the web.conf file. Required.
distsearch	tokenExchKeys	privateKeyPassphrase	This is needed for first-time-run and Splunk will not start without this. Required.
inputs	SSL	sslPassword	Password protecting the TOE's inputs server private key (indexer functionality). This is only used if Splunk is receiving data from a data feed on a TCP port using TLS. This was only used when Splunk was configured as an indexer in the evaluated configuration.
outputs	tcpout	sslPassword	Password protecting the TOE's outputs server private key (forwarder functionality). This is only used if Splunk is transmitting data to a data feed on a TCP port using TLS. This was only used when Splunk was configured as a forwarder in the evaluated configuration.
audit	auditTrail	privateKeyPassphrase	Audit-trail signing feature. Required for first-time-run.

The outputs.conf is only applicable for when the TOE is configured to use the forwarder functionality. The inputs.conf is only applicable for when the TOE is configured to use the indexer functionality. The alert_actions is only applicable when the TOE is configured to use the SMTP alert messaging functionality.

This data can be written to the keyring using the Splunk CLI with the following command: "secret-storage --write <conf> <stanza> <param>"

8.1.10 FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1:

The TOE implements TLS v1.2 in support of HTTPS/TLS secure communications.

The TOE acts as a TLS client for connections to the SMTP server and if Splunk is configured to transmit non-TSF data (i.e. forwarder functionality). As part of establishing a TLS connection as a client, the TSF will verify that the presented identifier in the peer certificate is a valid reference identifier.

The TOE acts as a TLS Server when configured to receive information (i.e. indexer functionality) from an external trusted data feed and for the administrative web GUI. When acting as a TLS server, the TSF will generate ECDHE over NIST curves secp256r1, secp384r1, and secp521r1 key establishment parameters. These are used for the key establishment process addressed by FCS_CKM.2. The TSF will also reject any TLS client request that is not using TLS v1.2. For external trusted data feed to TOE transfer, mutual authentication using client-side X.509v3 certificates is used to establish the TLS session.

The TOE will perform several TLS functions identically regardless of whether it is acting as a client or as a server. It will validate the peer certificate used for the connection. The TOE will not establish a trusted

channel if the peer certificate is invalid. Mutual authentication is supported and can be enabled/disabled administratively. The TOE does not support the use of URI names, Service names, IP addresses, wildcard certificates, or pinned certificates. The TOE can be configured within the .conf files to verify Common Name (CN) and/or Subject Alternative Names (SAN) presented identifiers. The CN hostname and SAN hostname (DNS name) are the only supported reference identifiers that can be forced as part of the certificate validation. In the evaluated configuration, the TSF is configured to support the following TLS cipher suites:

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

8.1.11 FCS_TLSC_EXT.2:

The application supports mutual TLS authentication using client-side X.509v3 certificates for TLS connections. The TOE will present its client certificate to the server upon request.

8.1.12 FCS_TLSC_EXT.4:

The supported Elliptic Curves Extension presented in the Client Hello include NIST curves secp256r1, secp384r1, and secp521r1. The NIST curves are available by default.

8.2 User Data Protection

8.2.1 FDP_DAR_EXT.1:

The TOE relies on the underlying platform to provide data-at-rest encryption. In addition to securely storing credential data in the GNOME keyring (see section 8.1.9 above), the private keys and filesystem objects that comprise the TOE itself can be stored on a drive partition that is secured using Linux Unified Key Setup (LUKS) encryption. Instructions for preparing the Operational Environment to allow for this are provided in the supplemental administrative guidance that is included with the TOE. The following table lists the data at rest that is secured by the Operational Environment:

Table 12: Data at Rest

Configuration Item	Stanza	Parameter	Description
Server	sslConfig	sslKeysFile	Encrypted private key + full certificate chain for splunkd server
		dhFile	DH parameter file
		sslRootCAPath	List of trusted root CAs (single .pem file)
	kvstore	sslKeysPath	Encrypted private key + full certificate chain for KVStore server
		sslCRLPath	CRL file for KVStore
Web	settings	privKeyPath	encrypted private key
		caCertPath	server cert
		dhFile	DH Param file
Distsearch	tokenExchKeys	privateKey publicKey	Encrypted private key

Inputs	SSL	serverCert	Full path to the server certificate for Splunk indexer functionality. Needed for receiving data sent by a data feed. This was only used when Splunk was configured as an indexer in the evaluated configuration.
		dhFile	DH Param file
Outputs	tcpout	sslCertPath	Full path to the client certificate on Splunk forwarder functionality. Needed for transmitting data to an external data feed. This was only used when Splunk was configured as a forwarder in the evaluated configuration.
$SPLUNK_ETC/auth/crl			CRL files used by Splunk must be stored in this directory

Security administrator credential data for the TOE is stored in the passwd file in $SPLUNK_ETC and protected using LUKS. Credentials can be overridden by a security administrator through configuration of the user-seed.conf file using the CLI, with the credential loaded into the GNOME keyring. Additionally, any CRL files used by the TOE must reside in the $SPLUNK_ETC/auth/crl directory.

8.2.2 **FDP_DEC_EXT.1:**

The TOE relies on its underlying platform to provide the actual network connectivity in order to establish communications channels. There are no sensitive information repositories (storage locations for private user or administrator data) that the TOE requires access to.

8.2.3 **FDP_NET_EXT.1:**

The TOE requires network access to facilitate remote administration. Additionally, the primary purpose of the product is to collect system generated data from multiple sources and apply various filters and formatting to that data. Therefore, the TOE indexer (TLS server) requires network access to receive non-TSF data from the operational environment data feed (i.e. Splunk forwarder). The TOE forwarder (TLS client) requires network access to transmit non-TSF data to the operational environment data feed receiver (i.e. Splunk indexer). In the evaluated configuration, the TOE indexer will also initiate the transmission of email alerts to a remote SMTP server (TLS client).

To support the administration of the TOE, the application initiates the default listening port 8000 for the web server to respond to remote administration requests (shows as the *splunkd* process name). Splunk also initiates the following default listening ports for internal Splunk processing support: 8089 (*splunkd*) management port and 8065 (*python*) application server which also support the administration of the TOE.

When the environment includes Splunk forwarders, a receiver port has to be established on the Splunk indexer. By default the Splunk indexer's receiver port uses port 9997 (*splunkd*). However, for the evaluated configuration port 9998 was used. The Splunk forwarder functionality does not use a default listening port as it is only transmitting data (TLS client) to the indexer.

8.3 Identification and Authentication

8.3.1 **FIA_X509_EXT.1:**

The TOE provides an internal mechanism to perform certificate validation. Specifically, the TSF performs the following checks in order to determine if a given certificate is valid:

- Certificate validation and certificate path validation conforms to RFC 5280.
- The certificate path must terminate with a trusted CA certificate.
- All CA certificates must have the basicConstraints extension present and the CA flag set to TRUE.
- The certificate must not be revoked. This is based on a certificate revocation list (CRL) that is consumed by the TOE at startup. CRL information can also be refreshed during runtime with the CLI command 'splunk reload crl'.
- The extendedKeyUsage field must be valid based on the following rules:
    - Certificates used for trusted updates and executable code integrity verification must have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS must have the Server Authentication purpose (id-kp with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS must have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - S/MIME certificates presented for email encryption and signature must have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses must have the OCSP signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
    - Server certificates presented for EST must have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field.
- The SAN/CN checks happens after all other certificate checks have happened (e.g. signature validation, expiry, certificate purpose etc.)
    - If SAN is defined in the .conf file:
        - If the SAN defined in presented certificate exactly matches the SAN defined in the .conf file, the certificate is accepted.
        - Otherwise certificate is rejected.
        - CN is not checked
    - If CN is defined in .conf file and SAN is not present in .conf file:
        - CN in presented certificate must match CN defined in .conf file.
        - If there are no CNs listed in the .conf file, the certificate is accepted.

8.3.2 **FIA_X509_EXT.2:**

The TOE uses X.509 certificates for HTTPS/TLS authentication. The use of certificates is enabled by default. However, a security administrator may configure the behavior of this function by specifying whether mutual authentication is supported. The security administrator may also specify the path to a certificate revocation list so that revocation status can be checked during authentication. The actual imported certificates and keys to be used by the TOE are specified through the use of .conf files. While

the HTTPS implementation will automatically reject a certificate if it is found to be invalid, a certificate with unknown revocation status (because the TSF is unable to read the CRL) is accepted.

8.4 Security Management

8.4.1 **FMT_CFG_EXT.1:**

The TOE requires credentials for remote administration via the web UI. The initial installation of the TOE prompts the security administrator to create a user name and password. Security administrators and users can only terminate their own interactive sessions. There are no default credentials.

The TOE runs as a non-root OS user 'splunk'. By default, the TOE installs with the following file permissions in its home directory (SPLUNK_HOME) and configuration directory (SPLUNK_ETC):

- The OS user ('splunk') has read-write-execute access
- The 'splunk' group has read-execute access
- No access is granted for 'other' users.

At startup, the TOE will ensure that 'other' users do not have any file access for SPLUNK_HOME and SPLUNK_ETC, overwriting file permissions if needed.

8.4.2 **FMT_MEC_EXT.1:**

The TOE is capable of using the underlying platform's recommended methods for storing and setting configuration options. In the TOE's evaluated configuration, all configuration information related to the Splunk application is stored in /etc/opt/splunk. This is done by specifying the SPLUNK_ETC environment variable to be the correct location (`export SPLUNK_ETC=/etc/opt/splunk`).

There are several dedicated configuration files with parameters and settings that are required for CC configuration. These configuration files include: server.conf (back-end communications between splunkd and Splunk Web), web.conf (remote web UI), alert_actions.conf (SMTP), inputs.conf (TLS server for indexer functionality), outputs.conf (TLS client for forwarder functionality). These parameters include the ability to configure the cipher suites, set the TLS version for both server and client communication, customize the reference identifier (SAN and CN), identify the X.509 certificate and storage path, enable certificate validation, and enable mutual authentication. See section 7.5 of the Splunk Enterprise 7.3 Supplemental Administrative Guidance for Common Criteria for the full description of parameter names and settings related to the SFRs and settings that are mandated for CC compliance.

8.4.3 **FMT_SMF.1:**

The TOE provides a remote web UI and local CLI to allow security administrators to manage the TSF. These interfaces provide functions that are related to the management of the Splunk application itself. Only the functions that are relevant to the TSF are discussed in this Security Target. The following security-relevant management functions are provided by the TOE:

- Ability to configure the supported TLS cipher suites.
- Ability to check the TOE version.

There is no management function for checking for TOE updates. The TOE automatically checks for

updates. The actual downloading and installation of any downloaded updates is manually performed by a person with root access to the platform and therefore is not included in this list.

8.5 Privacy

8.5.1 FPR_ANO_EXT.1:

The TOE does not collect personally identifiable information (PII) for security administrators or users. Therefore, there is no case in which the TOE will transmit this data over the network.

8.6 Protection of the TSF

8.6.1 FPT_AEX_EXT.1:

The TOE implements several mechanisms to protect against exploitation. Address Space Layout Randomization (ASLR) is enabled for the application itself (flags: -pie and -fPIE), which means that the TSF does not do any direct mapping of memory locations. The TOE does allocate memory regions with both write and execute permissions for the following functions only, both of which are used in support of just-in-time (JIT) compilation:

- sljit: used by Perl Compatible Regular Expressions (PCRE)
- libffi: used by Python to bind Python code to C code
- luajit: used by node.js to perform JIT compilation of Lua code.

The TOE also implements write/execute protection by not writing user-modifiable files to directories with executable files by default. The TOE is compatible with platform security features through the use of a SELinux profile that was created by the TOE developer. Additionally, the TOE was compiled using the -fstack-protector-strong compilation flag. The TOE itself will not replace, modify or replace its own executable code. Code updates can only be performed using the platform's package manager.

8.6.2 FPT_API_EXT.1:

Splunk Enterprise ships almost all of the libraries and scripting languages Splunk requires to operate and does not depend on the platform. Therefore, scripting languages like Python/Lua/JS are part of the TOE and are not platform APIs leveraged by TOE. Listed below are the only exceptions where Splunk leverages the platform's API (system calls).

Table 13: Platform APIs Used by the TOE

System Calls			
__assert_fail	fmod	mkstemp	setresgid
__ctype_b_loc	fopen	mkstemp64	setresuid
__ctype_get_mb_cur_max	fopen64	mktime	setreuid
__ctype_tolower_loc	fork	mmap	setrlimit
__ctype_toupper_loc	forkpty	mmap64	setrlimit64
__cxa_atexit	fpathconf	modf	setsid
__duplocale	fprintf	mprotect	setsockopt
__errno_location	fputc	msync	setuid
__fdelt_chk	fputs	munmap	setvbuf

__finite	fread	nanosleep	shutdown
__fprintf_chk	free	nftw	sigaction
__fread_chk	freeaddrinfo	nice	sigaddset
__freelocale	freeifaddrs	nl_langinfo	sigaltstack
__fxstat	frexp	open	sigemptyset
__fxstat64	fscanf	open64	sigfillset
__h_errno_location	fseek	opendir	siginterrupt
__isinf	fseeko	openpty	signal
__isinff	fseeko64	pathconf	sigpoll
__isnan	fstatfs	pause	sigprocmask
__isoc99_fscanf	fstatvfs64	pclose	sin
__isoc99_sscanf	fsync	perror	sincos
__libc_current_sigrtmax	ftell	pipe	sinh
__libc_current_sigrtmin	ftello	popen	sleep
__libc_start_main	ftello64	posix_fadvise	snprintf
__lxstat	ftruncate	posix_memalign	socket
__lxstat64	ftruncate64	pow	socketpair
__memcpy_chk	funlockfile	prctl	sprintf
__memmove_chk	fwrite	pread	sqrt
__memset_chk	gai_strerror	pread64	sqrtf
__newlocale	getaddrinfo	preadv64	srand
__nl_langinfo_l	getc	pthread_attr_destroy	srand48
__open64_2	getcwd	pthread_attr_init	sscanf
__pread64_chk	getdtablesize	pthread_attr_setscope	statfs
__printf_chk	getegid	pthread_attr_setstacksize	statvfs
__rawmemchr	getenv	pthread_barrier_destroy	statvfs64
__read_chk	geteuid	pthread_barrier_init	stderr
__realpath_chk	getgid	pthread_barrier_wait	stdin
__snprintf_chk	getgrent	pthread_cond_broadcast	stdout
__sprintf_chk	getgrgid_r	pthread_cond_destroy	stpcpy
__stack_chk_fail	getgrnam_r	pthread_cond_init	strcasecmp
__stpcpy_chk	getgroups	pthread_cond_signal	strcat
__strcat_chk	gethostbyaddr	pthread_cond_timedwait	strchr
__strcpy_chk	gethostbyname	pthread_cond_wait	strcmp
__strdup	gethostname	pthread_condattr_destroy	strcoll
__strncat_chk	getifaddrs	pthread_condattr_init	strcpy
__strncpy_chk	getitimer	pthread_create	strcspn
__sysv_signal	getloadavg	pthread_detach	strerror
__tls_get_addr	getlogin	pthread_getspecific	strftime
__uflow	getnameinfo	pthread_join	strftimel
__uselocale	getopt_long	pthread_key_create	strlen
__vfprintf_chk	getpagesize	pthread_key_create	strncasecmp
__vsnprintf_chk	getpeername	pthread_key_delete	strncat

__xmknod	getpgid	pthread_kill	strncmp
__xpg_strerror_r	getpgrp	pthread_mutex_destroy	strncpy
__xstat	getpid	pthread_mutex_init	strndup
__xstat64	getppid	pthread_mutex_lock	strrchr
abort	getpriority	pthread_mutex_trylock	strsignal
accept	getpwent	pthread_mutex_unlock	strspn
access	getpwnam	pthread_mutexattr_destroy	strstr
acos	getpwnam_r	pthread_mutexattr_init	strtod
alarm	getpwuid	pthread_mutexattr_settype	strtod
alphasort64	getpwuid_r	pthread_once	strtof
asctime_r	getresgid	pthread_rwlock_destroy	strtok
asin	getresuid	pthread_rwlock_init	strtol
atan	getrlimit	pthread_rwlock_rdlock	strtold
atan2	getrlimit64	pthread_rwlock_tryrdlock	strtoll
atoi	getrusage	pthread_rwlock_trywrlock	strtoul
backtrace	getservbyname	pthread_rwlock_unlock	strtoull
backtrace_symbols	getsid	pthread_rwlock_wrlock	strxfrm
bind	getsockname	pthread_self	symlink
bindtextdomain	getsockopt	pthread_setname_np	sync
btowc	gettext	pthread_setspecific	syscall
calloc	gettimeofday	pthread_sigmask	sysconf
ceil	getuid	putc	sysinfo
ceilf	getwc	putchar	system
cfmakeraw	gmtime_r	putenv	tan
chdir	hypot	puts	tanh
chmod	if_nametoindex	putwc	tcgetattr
chown	inet_addr	pwrite	tcgetpgrp
chroot	inet_aton	pwrite64	tcsetattr
clearerr	inet_ntoa	pwritev64	tcsetpgrp
clock	inet_ntop	qsort	tempnam
clock_getres	inet_pton	qsort_r	textdomain
clock_gettime	initgroups	raise	time
close	ioctl	rand	timegm
closedir	isalnum	read	times
confstr	isalpha	readdir	timespec_get
connect	isatty	readdir_r	tmpfile
cos	iscntrl	readdir64	tmpfile64
cosh	isgraph	readlink	tmpnam_r
ctermid	islower	readv	towlower
ctime	isprint	realloc	towupper
difftime	ispunct	recv	truncate
dirname	isspace	recvfrom	ttyname

dl_iterate_phdr	isupper	recvmsg	tzset
dladdr	iswctype	remove	umask
dlclose	isxdigit	rename	uname
dlerror	kill	rewind	ungetc
dlopen	killpg	rmdir	ungetwc
dlsym	lchown	round	unlink
dup	ldexp	scandir64	unsetenv
dup2	link	sched_yield	usleep
endgrent	listen	select	utime
endpwent	localeconv	sem_destroy	utimes
execv	localtime	sem_init	vsnprintf
execve	localtime_r	sem_post	wait
execvp	log	sem_timedwait	wait3
exit	log10	sem_trywait	wait4
exp	logf	sem_wait	waitpid
fchdir	lrand48	send	wcrtomb
fchmod	lrint	sendfile64	wcscmp
fchown	lseek	sendmsg	wcscoll
fclose	lseek64	sendto	wcsftime
fcntl	malloc	setegid	wcslen
fdatasync	mbrtowc	setenv	wcsnrtombs
fdopen	mbsnrtowcs	seteuid	wcsxfrm
feof	mbsrtowcs	setgid	wctod
ferror	memchr	setgroups	wctype
fesetround	memcmp	setitimer	wmemchr
fflush	memcpy	setlinebuf	wmemcmp
fgetc	memmove	setlocale	wmemcpy
fgets	memrchr	setpgid	wmemmove
fileno	memset	setpgrp	wmemset
flockfile	mkdir	setpriority	write
floor	mkdtemp	setpwent	writev
floorf	mkfifo	setregid	

8.6.3 **FPT_LIB_EXT.1:**

The TOE is package with several third-party open source libraries in order to function. The following is a list of the libraries used by the TOE:

Table 14: TOE Libraries

Component Name		
__m2crypto.so	libaep.so	libssl.so
_bisect.so	libarchive.so	libssl.so.1.0.0
_codecs_iso2022.so	libarchive.so.13	libsureware.so

_codecs_jp.so	libarchive.so.13.3.3	libubsec.so
_collections.so	libatalla.so	libxml2.so
_csv.so	libbson-1.0.so	libxml2.so.2
_elementtree.so	libbson-1.0.so.0	libxml2.so.2.9.9
_functools.so	libbson-1.0.so.0.0.0	libxmlsec1.so
_hashlib.so	libbz2.so	libxmlsec1.so.1
_heapq.so	libbz2.so.1	libxmlsec1.so.1.2.24
_io.so	libbz2.so.1.0.3	libxmlsec1-openssl.so
_json.so	libcapi.so	libxmlsec1-openssl.so.1
_locale.so	libchil.so	libxmlsec1-openssl.so.1.2.24
_multibytecodec.so	libcrypto.so	libxslt.so
_multiprocessing.so	libcrypto.so.1.0.0	libxslt.so.1
_random.so	libcswift.so	libxslt.so.1.1.30
_sha256.so	libexslt.so	libz.so
_sha512.so	libexslt.so.0	libz.so.1
_socket.so	libexslt.so.0.8.18	libz.so.1.2.11
_ssl.so	libgmp.so	math.so
_struct.so	libgost.so	objectify.so
array.so	libjemalloc.so	operator.so
binascii.so	libjemalloc.so.2	parser.so
bz2.so	libmongoc-1.0.so	pyexpat.so
cPickle.so	libmongoc-1.0.so.0	rand.so
crypto.so	libmongoc-1.0.so.0.0.0	resource.so
cStringIO.so	libnuron.so	select.so
datetime.so	libpadlock.so	SSL.so
etree.so	libpcre2-8.so	strop.so
fcntl.so	libpcre2-posix.so	termios.so
future_builtins.so	libsqlite3.so	time.so
itertools.so	libsqlite3.so.0	unicodedata.so
lib4758cca.so	libsqlite3.so.0.8.6	zlib.so

8.6.4 **FPT_TUD_EXT.1:**

The TOE provides the ability for security administrators to determine its currently installed version by using the Help→About in the web UI or through the underlying platform's package manager. The CLI command `splunk version` can also be used to query the current version of the TOE.

Splunk automatically checks to see if an update is available when a user is authenticated to the web UI. Splunk will notify the authenticated user with a message displayed on the post-authentication page, underneath the "Messages" menu if there is an update available. There is no update message presented to the authenticated user if there is no update available. Splunk does not download or install updates automatically.

After selecting the update URL, the user will be redirected to the authorized Splunk customer portal site where the customer must authenticate prior to being able must manually download the RPM package to

the underlying platform. This package must then be manually installed using the platform's RPM application by someone with root privilege. Splunk provides a public key within the RPM and is installed during the initial installation. The root administrator should run the "`rpm -K <filename.rpm>`" command which will verify the update against the installed public key prior to installation.

When removing the TOE from the platform, the package manager will erase the $SPLUNK_HOME directory where the TOE is originally installed. Any configuration settings or output files that were written to the /etc/opt/splunk directory, will be preserved in the /opt/splunk/etc directory structure. Log data is preserved under /opt/splunk/var/log and /opt/splunk/var/lib/splunk directory structure.

8.6.4.1 *Timely Security Updates*

As part of providing timely security updates, Splunk provides customers with a support section on splunk.com where they have the ability to submit support issues. This is an HTTPS site that requires user authentication prior to use. Any feedback that necessitates a fix will result in a patch to Splunk itself so there is no third-party update process to consider when updating the TOE. Any security fixes will be released as new packages in the same manner as any feature. Any implementation flaws are expected to be addressed within 90 days of reporting. Customers are notified of security-related fixes on the Splunk customer portal.

8.7 Trusted Path/Channel

8.7.1 **FTP_DIT_EXT.1:**

The TOE uses HTTPS and TLS v1.2 to secure data in transit over trusted channels and paths. The TOE acts as an HTTPS server for remote administration performed using the web UI.

The TOE indexer also transmits smtp alert notifications to an SMTP server in the operational environment so that security administrators can receive email notifications of certain events. This communications channel is secured using TLS. Once the messages reach the SMTP server in the operational environment, any subsequent security of the email transmissions themselves (such as S/MIME) cannot be enforced using the TSF. For receiving and transmitting non-TSF data to and from an external trusted entity, the security is achieved using TLS v1.2 (HTTPS/TLS) with the TOE indexer acting as the server and the TOE forwarder as a client.