Splunk Enterprise 6.4.5

Security Target

ST Version: 1.0
January 20, 2017

Splunk
250 Brannan Street
San Francisco, CA 94107

Prepared By:

Booz | Allen | Hamilton
delivering results that endure

Cyber Assurance Testing Laboratory
304 Sentinel Drive
Annapolis Junction, MD 20701

Table of Contents

1 Security Target Introduction 5
1.1 ST Reference 5
1.1.1 ST Identification 5
1.1.2 Document Organization 5
1.1.3 Terminology 6
1.1.4 Acronyms 6
1.2 TOE Reference 7
1.3 TOE Overview 7
1.4 TOE Type 8
2 TOE Description 10
2.1 Evaluated Components of the TOE 10
2.2 Components and Applications in the Operational Environment 10
2.3 Excluded from the TOE 10
2.3.1 Not Installed 10
2.3.2 Installed but Requires a Separate License 10
2.3.3 Installed But Not Part of the TSF 11
2.4 Physical Boundary 11
2.4.1 Hardware 11
2.4.2 Software 11
2.5 Logical Boundary 11
2.5.1 Cryptographic Support 11
2.5.2 User Data Protection 12
2.5.3 Identification and Authentication 12
2.5.4 Security Management 12
2.5.5 Privacy 12
2.5.6 Protection of the TSF 12
2.5.7 Trusted Path/Channels 13
3 Conformance Claims 14
3.1 CC Version 14
3.2 CC Part 2 Conformance Claims 14

3.3 CC Part 3 Conformance Claims 14
3.4 PP Claims 14
3.5 Package Claims 14
3.6 Package Name Conformant or Package Name Augmented 15
3.7 Conformance Claim Rationale 15
4 Security Problem Definition 16
4.1 Threats 16
4.2 Organizational Security Policies 16
4.3 Assumptions 16
4.4 Security Objectives 16
4.4.1 TOE Security Objectives 16
4.4.2 Security Objectives for the Operational Environment 17
4.5 Security Problem Definition Rationale 17
5 Extended Components Definition 19
5.1 Extended Security Functional Requirements 19
5.2 Extended Security Assurance Requirements 19
6 Security Functional Requirements 20
6.1 Conventions 20
6.2 Security Functional Requirements Summary 20
6.3 Security Functional Requirements 22
6.3.1 Class FCS: Cryptographic Support 22
6.3.2 Class FDP: User Data Protection 24
6.3.3 Class FIA: Identification and Authentication 25
6.3.4 Class FMT: Security Management 26
6.3.5 Class FPR: Privacy 26
6.3.6 Class FPT: Protection of the TSF 26
6.3.7 Class FTP: Trusted Path/Channels 27
6.4 Statement of Security Functional Requirements Consistency 27
7 Security Assurance Requirements 28
8 TOE Summary Specification 29
8.1 Timely Security Updates 29

8.2 Cryptographic Support 29
8.2.1 FCS_CKM_EXT.1: 29
8.2.2 FCS_CKM.2: 29
8.2.3 FCS_COP.1(1): 29
8.2.4 FCS_COP.1(2): 29
8.2.5 FCS_COP.1(3): 29
8.2.6 FCS_COP.1(4): 30
8.2.7 FCS_HTTPS_EXT.1: 30
8.2.8 FCS_RBG_EXT.1 and FCS_RBG_EXT.2: 30
8.2.9 FCS_STO_EXT.1: 30
8.2.10 FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1: 31
8.3 User Data Protection 32
8.3.1 FDP_DAR_EXT.1: 32
8.3.2 FDP_DEC_EXT.1: 33
8.3.3 FDP_NET_EXT.1: 33
8.4 Identification and Authentication 33
8.4.1 FIA_X509_EXT.1: 33
8.4.2 FIA_X509_EXT.2: 34
8.5 Security Management 34
8.5.1 FMT_CFG_EXT.1: 34
8.5.2 FMT_MEC_EXT.1: 35
8.5.3 FMT_SMF.1: 35
8.6 Privacy 35
8.6.1 FPR_ANO_EXT.1: 35
8.7 Protection of the TSF 35
8.7.1 FPT_AEX_EXT.1: 35
8.7.2 FPT_API_EXT.1: 36
8.7.3 FPT_LIB_EXT.1: 36
8.7.4 FPT_TUD_EXT.1: 42
8.8 Trusted Path/Channels 42
8.8.1 FTP_DIT_EXT.1: 42

Appendix A: Platform APIs Supported by the TOE 44

Table of Figures

Figure 1-1: TOE Boundary 7

Table of Tables

Table 1-1: Customer Specific Terminology 6
Table 1-2: Acronym Definition 6
Table 2-2: Evaluated Components of the Operational Environment 10
Table 4-1: TOE Threats 16
Table 4-2: TOE Assumptions 16
Table 4-3: TOE Objectives 17
Table 4-4: TOE Operational Environment Objectives 17
Table 6-1: Security Functional Requirements for the TOE 21
Table 8-1: Credentials Stored in Keyring 31
Table 8-2: Data at Rest 33
Table 8-3: TOE Libraries 42

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 6.4.5 Security Target
ST Version: 1.0
ST Publication Date: January 20, 2017
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 1-1. 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.

Term	Definition
Administrator	An 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 GUI") is a web-based application that can be used to manage the TOE remotely using HTTPS.
User	An individual who has access to the TOE but is not able to manage its behavior.

Table 1-1: Customer Specific Terminology

1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 1-2. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
AES	Advanced Encryption Standard
ASLR	Address Space Layout Randomization
CA	Certification Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CLI	Command Line Interface
CRL	Certificate Revocation List
CSP	Critical Security Parameter
DHE	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
HMAC	Hashed Message Authentication Code
HTTPS	Hypertext Transfer Protocol - Secure
JIT	Just-in-Time (compilation)
PCRE	Perl Compatible Regular Expressions
RA	Registration Authority
RSA	Rivest Shamir Adelman (encryption algorithm)
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SMTP	Simple Mail Transfer Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function

Table 1-2: Acronym Definition

1.2 TOE Reference

The TOE is Splunk Enterprise 6.4.5, which is an application on an operating system. In its evaluated configuration, the TOE is a self-contained instance of Splunk Enterprise 6.4.5. In the evaluated configuration, there will be two or more instances of Splunk Enterprise 6.4.5 deployed and communicating with each other. One instance serves as an indexer that is responsible for aggregating non-TSF system generated data and one or more instances are configured as a forwarder that is responsible for collecting non-TSF system generated data on its underlying OS platform.

1.3 TOE Overview

The primary functionality of Splunk Enterprise 6.4.5 ("Splunk") is to collect system generated IT data from various types of OS platform systems and aggregate it in a centralized location for real-time visibility and analysis of system behavior. Splunk can be deployed in a distributed manner so that multiple instances of the product can act as forwarders, each of which is responsible for sending its data to a centralized instance of the product that is responsible for indexing the data and providing search/reporting capability for it via administrative interfaces. The Target of Evaluation (TOE) is the Splunk application itself, the interfaces used to administer it, and the communications channel that is used to transmit data between forwarder system(s) and the indexer. The general application security functionality of Splunk is the product functionality that is being evaluated as part of the TOE.

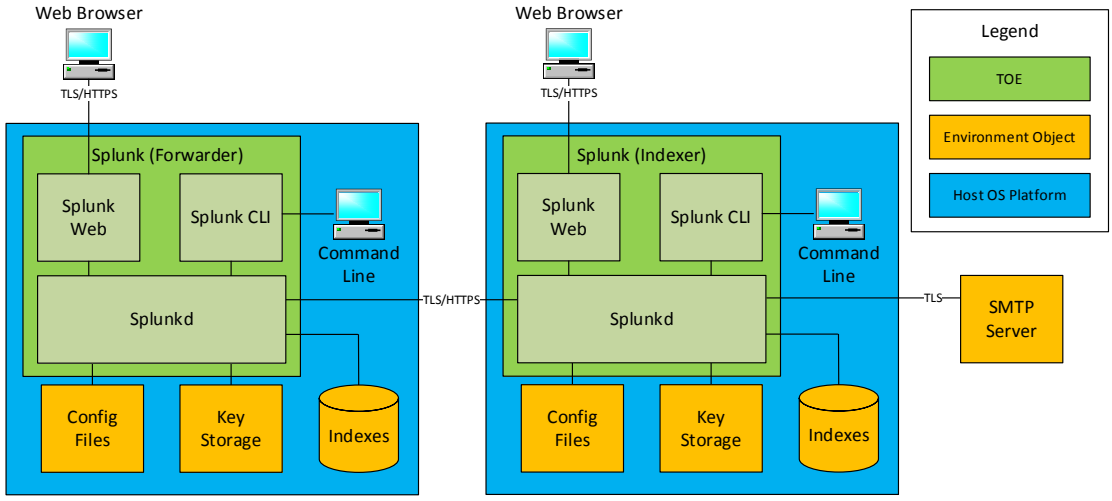

Figure 1-1: TOE Boundary

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, respectively. Splunkd is the application process that provides most of the product functionality.

Splunk can act as either a forwarder (responsible for collecting system generated data from the general-purpose computer that it resides on) or an indexer (responsible for receiving system generated data from one or more other instances of Splunk). Regardless of how Splunk is configured to function, the underlying binary application is the same; there is no separate executable code. While the product vendor provides multiple versions of the product, only the full version of Splunk Enterprise 6.4.5 is considered to be the TOE – other product versions were not evaluated and no security claims are made for them.

When the Splunk application is administratively configured as a forwarder, the administrator may specify an indexer application to transmit the system generated data that it collects to a centralized location. This transmission is secured using TLS/HTTPS. When the Splunk application is administratively configured as an indexer, the TOE can connect with an SMTP server to send out configured alerts based on the indexer's analysis of the system generated data that it receives.

Splunk Web is a browser-based UI that supports Internet Explorer 11 and the latest versions of Google Chrome and Mozilla Firefox. It provides a graphical interface to the product in order to perform 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.

Splunk CLI provides a command line interface to the product that can be accessed 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. For instance, a user would enter "splunk stop" to stop the Splunkd process.

Since Splunk is a standalone application that runs on a general-purpose computer (i.e. user's workstation), a single instance of the product is managed individually. If multiple instances of Splunk are configured as forwarders and transmit data to a Splunk instance configured as an indexer as defined by the evaluated configuration, this would be specified during initial setup of each Splunk instance as well as any further management. Thus, the multiple instances of Splunk in the evaluated configuration are not centrally managed as a distributed product.

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 IT 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.

Instances of Splunk that are configured as forwarders are installed on the systems from which logs will be collected and sent to the indexer. The sole function and communication pathway of the forwarders in the single instance configuration is to do this log collection and transfer to the indexer.

1.4 TOE Type

The TOE type for Splunk Enterprise 6.4.5 is Application Software. The Protection Profile for Application Software specifies several use cases that conformant TOEs may implement. In particular, 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 is considered to implement content consumption

because it allows a user to consume log data stored on the local filesystem 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 Splunk Enterprise 6.4.5 ("Splunk"), which includes the Splunkd process and the Splunk Web and Splunk CLI administrative interfaces. The TOE is a self-contained instance of Splunk Enterprise 6.4.5. In the evaluated configuration, the TOE's deployment will include a distributed collection of Splunk instances deployed on two or more host platforms. One instance of Splunk is configured as an indexer, which allows it to act as a server for receiving information from other instances of Splunk. One or more instances of Splunk are configured as a forwarder, which act as a client when sending data to the indexer.

2.2 Components and Applications in the Operational Environment

The following table lists components and applications in the environment that the TOE relies upon in order to function properly:

Component	Definition
Host Platform(s)	A general-purpose computer on which the TOE is installed. In the evaluated configuration, at least two host platforms are used.
Management Workstation	Any general-purpose computer that is used by an 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 distribute them to users in the Operational Environment via email.

Table 2-1: Evaluated Components of the Operational Environment

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 6.4.5 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 configured to only permit HTTPS remote communications.

Additionally, the TOE includes a number of functions that are outside the scope of the claimed Protection Profile. These functions are not part of the TSF because there are no SFRs that apply to them.

2.4 Physical Boundary

2.4.1 Hardware

Splunk Enterprise 6.4.5 is a software-only TOE and therefore has system requirements for the host platform that it is installed on. The following are the minimum system requirements for the evaluated configuration of the TOE:

- Red Hat Enterprise Linux 6.5, 64 bit
- 2x six-core, 2 GHz CPU (Intel Xeon x64)
- 12 GB RAM
- RAID 0 or 1+0
- 5 GB of free disk space

Note that it is recommended to avoid installing Splunk on an NFS file system for performance reasons.

2.4.2 Software

The physical boundary of the TOE software is the Splunk application, which includes the Splunkd application process, the Splunk Web administrative GUI, and the Splunk CLI executable command-line interface. The TOE software includes the OpenSSL FIPS Object Module v2.0.2 which performs the TOE's cryptographic operations.

2.5 Logical Boundary

The TOE is comprised of several security features. Each of the security features identified above consists of several security functionalities, as identified below.

1. Cryptographic Support
2. User Data Protection
3. Identification and Authentication
4. Security Management
5. Privacy
6. Protection of the TSF
7. Trusted Path/Channels

2.5.1 Cryptographic Support

The TOE uses NIST-validated cryptographic algorithm implementations, provided by OpenSSL FIPS

Object Module v2.0.2, to support the establishment of trusted channels and paths to protect data in transit. In the evaluated configuration, the TOE will act as a server for TLS/HTTPS to facilitate trusted remote communications. 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).

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 will only use the underlying platform's network connectivity for administrative activities, email alerts that are generated with user approval, and administratively-configured transmission of system generated data from a forwarder instance of Splunk to an indexer.

2.5.3 Identification and Authentication

In order to facilitate secure communications using HTTPS, the TOE provides a mechanism to validate X.509 certificates. The TOE uses a CRL to check certificate expiration status but will permit certificates to be used (with a warning) when the CRL is unavailable.

2.5.4 Security Management

The TOE provides a default credential that is used for initial authentication that must be reset prior to any other TSF-mediated action being authorized. 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 provides several security-relevant management functions. Specifically, the TOE has the ability to configure how information about the underlying platform is transmitted over the network. The TOE also provides administrators with the ability to configure the behavior of the TLS/HTTPS trusted channel. Any changes to the TOE's configuration will be logged.

2.5.5 Privacy

The TOE ensures the privacy of its administrators and users by not providing any ability 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 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/Channels

The TOE protects all data in transit using HTTPS over TLS or standalone TLS. TLS/HTTPS protocol is used to secure remote administration using the web GUI. It can also be used to securely send alerts to a remote SMTP server in the Operational Environment. TLS is used to secure communications between separate instances of Splunk, where the forwarder instance(s) act as the client and the indexer acts as a server.

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 20, 2017.

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 20, 2017.

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.

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

- FCS_CKM_EXT.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.1
- FCS_RBG_EXT.2
- FCS_TLSC_EXT.1
- FCS_TLSS_EXT.1
- FIA_X509_EXT.1
- FIA_X509_EXT.2

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

3.6 Package Name Conformant or Package Name Augmented

This ST and TOE are conformant with the claimed 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.

4 Security Problem Definition

4.1 Threats

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

Threat	Threat Definition
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.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.PHYSICAL_ACCESS	An attacker may try to access sensitive data at rest.

Table 4-1: TOE Threats

4.2 Organizational Security Policies

The App PP defines no organizational security policies.

4.3 Assumptions

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

Assumption	Assumption Definition
A.PLATFORM	The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.
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.
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.

Table 4-2: TOE Assumptions

4.4 Security Objectives

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

4.4.1 TOE Security Objectives

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

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.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.
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.
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.
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.

Table 4-3: TOE Objectives

4.4.2 Security Objectives for the Operational Environment

The TOE's operating environment must satisfy the following objectives, as defined by the App PP:

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_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.
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.

Table 4-4: TOE Operational Environment Objectives

4.5 Security Problem Definition Rationale

The assumptions, threats, OSPs, and objectives that are defined in this ST represent the assumptions, threats, OSPs, and objectives that are specified in the Protection Profile to which the TOE claims

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

5 Extended Components Definition

5.1 Extended Security Functional Requirements

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

5.2 Extended Security Assurance Requirements

The extended Security Assurance Requirement that is claimed in this ST is taken directly from the PP to which the ST and TOE claim conformance. This extended component is formally defined in the PP in which its usage is required.

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:

Class Name	Component Identification	Component Name
Cryptographic Support	FCS_CKM_EXT.1	Cryptographic Key Generation Services
	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 Secrets
	FCS_TLSC_EXT.1	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

Class Name	Component Identification	Component Name
	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 /Channels	FTP_DIT_EXT.1	Protection of Data in Transit

Table 6-1: Security Functional Requirements for the TOE

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 [generate no asymmetric cryptographic keys].

6.3.1.2 *FCS_CKM.2 Cryptographic Key Establishment*

FCS_CKM.2.1 The application shall [implement functionality] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:

[RSA-based key establishment schemes] that meets the following: [NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"] and [no other schemes].

6.3.1.3 *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 80038A) mode;

and [no other modes]

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

6.3.1.4 *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-1, SHA-256] and message digest sizes [160, 256] bits that meet the following: FIPS Pub 180-4.

6.3.1.5 *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

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

6.3.1.6 *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 [selection: SHA-1] with key sizes [*equal to block sizes*] and message digest sizes 256 and [160] bits that meet the following: FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard.

6.3.1.7 *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 The application shall implement HTTPS using TLS (FCS_TLSC_EXT.1).

FCS_HTTPS_EXT.1.3 The application shall notify the user and [not establish the connection] if the peer certificate is deemed invalid.

6.3.1.8 *FCS_RBG_EXT.1 Random Bit Generation Services*

FCS_RBG_EXT.1.1 The application shall [implement DRBG functionality] for its cryptographic operations.

6.3.1.9 *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.

6.3.1.10 *FCS_STO_EXT.1 Storage of Secrets*

FCS_STO_EXT.1.1 The application shall [invoke the functionality provided by the platform to securely store [*credentials defined in Table 8-1*]] to nonvolatile memory.

6.3.1.11 *FCS_TLSC_EXT.1 TLS Client Protocol*

FCS_TLSC_EXT.1.1 The application shall [implement TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

Mandatory Ciphersuites: TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246

Optional Ciphersuites: [

TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,

TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246].

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.

6.3.1.12 *FCS_TLSS_EXT.1 TLS Server Protocol*

FCS_TLSS_EXT.1.1 The application shall [implement TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

Mandatory Ciphersuites: TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246

Optional Ciphersuites: [

TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,

TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246].

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 The application shall generate key establishment parameters using RSA with size 2048 bits and [no other sizes] and [no other].

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 The application shall [leverage platform-provided functionality to encrypt sensitive data] in non-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 [user-initiated communication for [*remote administration*], [*transmission of alerts to environmental SMTP server, transmission of system generated data from Splunk forwarder to Splunk indexer*]].

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 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 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 The application shall be configured by default with file permissions which protect it and its data from unauthorized access.

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.

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 the transmission of any information describing the system's hardware, software, or configuration, [*enable/disable supported TLS ciphersuites, enable/disable TLS mutual authentication, 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 [*python2.7*].

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 8-3*].

6.3.6.4 *FPT_TUD_EXT.1 Trusted 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/Channels

6.3.7.1 *FTP_DIT_EXT.1 Protection of Data in Transit*

FTP_DIT_EXT.1.1 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.

7 Security Assurance Requirements

Because the ST and TOE claim exact conformance to the App PP, the Security Assurance Requirements (SARs) that are claimed are identical to those defined in the claimed PP and have not been reproduced.

8 TOE Summary Specification

8.1 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 updates (see discussion on FPT_TUD_EXT.1 below). The TOE contains a number of components, including third party components that Splunk does not have control over the implementation of. 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.

8.2 Cryptographic Support

8.2.1 FCS_CKM_EXT.1:

While the TOE uses asymmetric cryptography in support of TLS/HTTPS trusted communications, the asymmetric keys used for this functionality are imported from the Operational Environment. In the evaluated configuration, the TOE's web server, inter-TOE transfer mechanism, and any remote administrators or SMTP servers will be assigned certificates that are generated prior to installation of the TOE. Therefore, the TOE does not generate asymmetric keys nor does it invoke an environmental mechanism to do so.

8.2.2 FCS_CKM.2:

The TOE supports RSA key establishment schemes for establishment of TLS/HTTPS communications. RSA key establishment conforms to NIST SP 800-56B. This function is vendor-asserted under FIPS 140-2 IG D.4, Vendor Affirmation.

8.2.3 FCS_COP.1(1):

The TOE performs AES encryption in support of TLS/HTTPS communications. AES-CBC is the supported mode and both 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 number #2234.

8.2.4 FCS_COP.1(2):

The TOE performs cryptographic hashing in support of TLS/HTTPS. Both SHA-1 and SHA-256 are supported. This functionality is handled by the TOE's OpenSSL cryptographic module and has been certified by CAVP under SHS certificate number #1923.

8.2.5 FCS_COP.1(3):

The TOE provides digital signature services in support of validation of software updates and X.509v3 authentication. The TOE supports RSA signatures with sizes of 2048 bits or greater. This functionality is

handled by the TOE's OpenSSL cryptographic module and has been certified by CAVP under RSA certificate number #2205.

8.2.6 FCS_COP.1(4):

The TOE provides keyed-hash message authentication services in support of TLS/HTTPS communications. The TOE supports both HMAC-SHA-1 and HMAC-SHA-256 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 number #1363.

8.2.7 FCS_HTTPS_EXT.1:

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

- Remote administrator to TOE (via Web GUI)
- TOE to SMTP Server

Both interfaces use the same HTTPS implementation, which relies on the TLS client and server capability that is described in section 8.2.10. 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 8.4.2.

8.2.8 FCS_RBG_EXT.1 and FCS_RBG_EXT.2:

The TOE implements its own DRBG functionality but collects entropy from the underlying platform as a seed. The TOE includes an OpenSSL implementation of the AES_CTR DRBG (CAVP DRBG certificate #264) 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 TOE's DRBG is seeded with entropy data that is collected from /dev/urandom on the underlying platform. 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.

8.2.9 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 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:

Configuration Item	Stanza	Parameter	Description
passwords	credential	password	Credential information used for a given connecting app in Splunk Enterprise. Not required for CC Configuration.
alert_actions	email	auth_password	SMTP password for sending email. This is only used if Splunk is configured to send email alerts.
server	general	pass4SymmKey	Symmetric key used in licensing. Not required for CC Configuration.
server	sslConfig	sslKeysfilePassword	Passphrase protecting splunkd server private key. Required.
server	kvstore	sslKeysPassword	Passphrase protecting key-value store private key. Required.
user-seed	user_info	PASSWORD	Password for the seed user when Splunk does not have any existing users. This is only used if Splunk is being configured with the user-seed.conf file.
web	settings	privKeyPassword	Passphrase protecting the GUI web server private key. This is only used if Splunk is being configured to use the Web GUI 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	password	Password protecting the indexer's inputs server private key. This is only used if Splunk is accepting data from a forwarder on a TCP port using SSL.
outputs	tcpout	sslPassword	Password protecting the private key of outbound connection from forwarder to indexer using SSL.
audit	auditTrail	privateKeyPassphrase	Audit-trail signing feature. Required for first-time-run.

Table 8-1: Credentials Stored in Keyring

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

8.2.10 FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1:

The TOE implements TLS 1.2 in support of HTTPS secure communications as well as intra-TOE transfer of system generated data collected by an instance of Splunk acting as a forwarder. 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.

When acting as a TLS server, the TSF will generate 2048-bit RSA 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 1.2. For intra-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. Mutual authentication is supported and can be enabled/disabled administratively. The TOE does not support the use of Wildcards or IP Addresses. The TOE can be configured within the .conf files to verify Common Name (CN) and/or Subject Alternative Names (SAN) reference identifiers. The CN and SAN are the only supported reference identifiers that can be forced as part of the certificate validation. The TSF supports the following TLS ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256

8.3 User Data Protection

8.3.1 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 8.2.9 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:

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 dhFile	Full path to the server certificate on indexer for accepting data sent by forwarders DH Param file

outputs	tcpout	sslCertPath	Full path to the client certificate on forwarders.
$SPLUNK_ETC/auth/crl			CRL files used by Splunk must be stored in this directory

Table 8-2: Data at Rest

Administrator credential data for the TOE is stored in the passwd file in $SPLUNK_ETC and protected using LUKS, with the exception of the default credential. The default credential can be overridden with an administrator-specified value through configuration of the user-seed.conf file, 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.

8.3.2 FDP_DEC_EXT.1:

The TOE relies on its underlying platform to provide network connectivity. The supplemental administrative guidance documentation that accompanies the TOE provides the administrator with this information. There are no sensitive information repositories (in the sense of storage locations for private user or administrator data) that the TOE requires access to.

8.3.3 FDP_NET_EXT.1:

The TOE requires network access to facilitate remote administration. Since the primary purpose of the product itself is to collect system generated data from the platform and apply various filters and formatting to that data, the TOE also requires network access to transmit information related to this data, specifically in the cases of transmitting collected data from a forwarder application to an indexer application and of transmitting alerts to a remote SMTP server based on some observed behavior.

8.4 Identification and Authentication

8.4.1 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, we accept the cert.

8.4.2 FIA_X509_EXT.2:

The TOE uses X.509 certificates for TLS/HTTPS authentication. The use of certificates is enabled by default. However, an administrator may configure the behavior of this function by specifying whether mutual authentication is supported. The administrator may also specify the path to a certificate revocation list so that revocation status can be checked during authentication. The actual certificates and keys to be used by the TOE can be 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. In this case, the administrator is warned that the revocation check could not be performed.

8.5 Security Management

8.5.1 FMT_CFG_EXT.1:

The TOE requires credentials for remote administration via the Web GUI. The initial installation of the TOE creates an account with default credentials that is used for initial login. Once these credentials have been provided, the administrator is prompted to change the credentials before any other administrative actions can be performed. If for some reason all administrator accounts are eliminated, the TOE will re-instantiate the default credential so that administrators are not permanently locked out of the interface. Administrators and users can only terminate their own interactive sessions.

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.

8.5.2 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.

8.5.3 FMT_SMF.1:

The TOE provides a remote web GUI and local CLI to allow administrators to manage the TSF. These interfaces provide a large number of 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 enable/disable the transmission of any information describing the system’s hardware, software, or configuration – specifically, this is done by configuring email alerts about system activity that the TOE can send.
- Ability to enable/disable the TOE’s TLS mutual authentication implementation.
- Ability to configure the supported TLS ciphersuites.
- 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.

8.6 Privacy

8.6.1 FPR_ANO_EXT.1:

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

8.7 Protection of the TSF

8.7.1 FPT_AEX_EXT.1:

The TOE implements several mechanisms to protect against exploitation. Address Space Layout Randomization (ASLR) is enabled for the application itself, 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.

8.7.2 FPT_API_EXT.1:

The TOE uses only supported platform APIs in order to function. The APIs used by the TOE are listed in Appendix A.

8.7.3 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:

Component Name	Component Version Number	Main Resource Location	Component Functionality
Highcharts	4.0.4	www.highcharts.com	Javascript Charting Library
MaxMind GeoLite-City	N/A	http://www.maxmind.com/en/builder	MaxMind database that contains IP to Geolocation information.
MongoDB	v3.0.8-splunk (83d8cc25e00e42856924d84e220fbe4a839e605d)	https://www.mongodb.com/	Database
Sound Effects from audioblocks.com	N/A	audioblocks.com	3 audio files to be played when a user hits a threshold of keyword matches.
Component Name	**Component Version Number**	**Main Resource Location**	**Component Functionality**
Aaargh (Dated: 10/17/12)	0.4	https://github.com/wbolster/aaargh	Aaargh is a Python module that makes building friendly command line applications really easy.
Ace	1.2.2	https://ace.c9.io	Ace is a standalone code editor written in JavaScript.
almond.js	0.2.9	http://github.com/jrburke/almond	almond.js is a replacement AMD loader for RequireJS. It provides a minimal AMD API footprint that includes loader plugin support. Only useful for built/bundled AMD modules, does not do dynamic loading.
Apache Common IO	2.4	https://commons.apache.org/proper/commons-io/download_io.cgi	Commons IO is a library of utilities to assist with developing IO functionality.
Apache Commons Codec	1.6	https://commons.apache.org/proper/commons-codec/	Apache Commons Codec (TM) software provides implementations of common encoders and decoders such as Base64, Hex, Phonetic and URLs.
Apache Commons Compress	1.10	https://commons.apache.org/proper/commons-compress/index.html	The Apache Commons Compress library defines an API for working with ar, cpio, Unix dump, tar, zip, gzip, XZ, Pack200, bzip2, 7z, arj, lzma, snappy, DEFLATE and Z files.
Apache Hive	0.12.0	https://hive.apache.org/index.html	The Apache Hive ♦ data warehouse software facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL. At the same time this language also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL.
Apache Hive Metastore	0.12.0	https://cwiki.apache.org/confluence/display/Hive/AdminManual+MetastoreAdmin	All the metadata for Hive tables and partitions are accessed through the Hive Metastore. Metadata is persisted using JPOX ORM solution (Data Nucleus) so any database that is

			supported by it can be used by Hive. Most of the commercial relational databases and many open source databases are supported.
Apache Hive Serde	0.12.0	https://hive.apache.org/javadocs/r0.12.0/api/org/apache/hadoop/hive/serde2/SerDe.html	A union of HiveDeserializer and HiveSerializer interface. If a developer wants his hive table to be read-only, then he just want to return both readable and writable, then All serdes should extend the abstract class AbstractSerDe, and eventually SerDe interface should be removed
Apache Parquet	1.5.0	http://parquet.apache.org/	Apache Parquet is a columnar storage format available to any project in the Hadoop ecosystem, regardless of the choice of data processing framework, data model or programming language.
Apache Thrift	0.9.2	https://thrift.apache.org/	The Apache Thrift software framework, for scalable cross-language services development, combines a software stack with a code generation engine to build services that work efficiently and seamlessly between C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, JavaScript, Node.js, Smalltalk, OCaml and Delphi and other languages.
Avro	1.7.7	https://avro.apache.org/releases.html	Apache Avro&#65533; is a data serialization system. Avro provides: Rich data structures. A compact, fast, binary data format. A container file, to store persistent data. Remote procedure call (RPC). Simple integration with dynamic languages. Code generation is not required to read or write data files nor to use or implement RPC protocols. Code generation as an optional optimization, only worth implementing for statically typed languages.
Avro MapReduce	1.7.7	https://avro.apache.org/docs/1.7.7/api/java/org/apache/avro/mapred/package-summary.html	Run Hadoop MapReduce jobs over Avro data, with map and reduce functions written in Java.
AWS Java SDK	1.10.8	http://aws.amazon.com/sdk-for-java/	The SDK helps take the complexity out of coding by providing Java APIs for many AWS services including Amazon S3, Amazon EC2, DynamoDB, and more.
Babel	0.9.4	http://babel.edgewall.org	Babel is a Python library that provides an integrated collection of utilities that assist with internationalizing and localizing Python applications (in particular web-based applications.)
Backbone.js	1.1.2	http://backbonejs.org	Backbone.js gives structure to web applications by providing models with key-value binding and custom events, collections with a rich API of enumerable functions, views with declarative event handling, and connects it all to your existing API over a RESTful JSON interface.
Backbone.validation	0.9.1	http://thedersen.com/projects/backbone-validation/	A validation plugin for Backbone.js that validates both your model as well as form input
Beaker	1.6.5	http://beaker.groovie.org	Beaker is a library for caching and sessions for use with web applications and stand-alone Python scripts and applications.
bindings	1.2.1	https://github.com/TooTallNate/node-bindings/	Helper module for loading your native module's .node file This is a helper module for authors of Node.js native addon modules. It is basically the "swiss army knife" of require()ing your native module's .node file.
blanket.js	1.1.5	http://blanketjs.org	Blanket.js is a JavaScript code coverage library that works both in-browser and with nodejs.
Bootstrap	2.3.1	http://blog.getbootstrap.com/2013/03/01/bootstrap-2-3-1-released/	Provides extensive documentation for common HTML elements, dozens of custom HTML and CSS components, and awesome jQuery plugins.
bottle	0.12.8	bottlepy.org	A small webserver for DMC
bzip2	1.0.6	http://bzip.org	bzip2 is a high-quality data compressor
CherryPy	3.1.2	http://www.cherrypy.org	CherryPy is a Python web framework that allows developers to build web applications
CodeMirror	2.38	http://codemirror.net/	CodeMirror is a versatile text editor implemented in JavaScript for the browser
CoffeeScript	1.1.2 (Dated: 8/4/11)	http://coffeescript.org/	CoffeeScript is a language that compiles into JavaScript.
Contextify	0.1.9	https://github.com/brianmcd/contextify	Contextify turns an object into a V8 execution context. A contextified object acts as the global 'this' when executing scripts in its context.

Cookies	0.3.0 (Dated: 4/26/12)	https://github.com/jed/cookies	Cookies is a node.js module for getting and setting HTTP(S) cookies
cssmin	0.1.4 (Dated: 5/5/10)	https://github.com/zacharyvoase/cssmin	A Python port of the YUI CSS compression algorithm.
CSSOM	0.3.0	https://github.com/NV/CSSOM	CSSOM.js is a CSS parser written in pure JavaScript
cssstyle	0.2.21	https://github.com/chad3814/CSSStyleDeclaration	CSSStyleDeclaration is a work-a-like to the CSSStyleDeclaration class in Nikita Vasilyev's CSSOM. I made it so that when using jQuery in node setting css attributes via $.fn.css() would work. node-jquery uses jsdom to create a DOM to use in node. jsdom uses CSSOM for styling, and CSSOM's implementation of the CSSStyleDeclaration doesn't support CSS2Properties, which is how jQuery's $.fn.css() operates.
d3.js	3.4.13	http://d3js.org/	D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.
d3.v2.js	2.10.2	http://d3js.org/	D3.js is a JavaScript library for manipulating documents based on data
decorator	2.2.0 (Dated:7/31/07)	http://pypi.python.org/pypi/decorator	Python decorator library
Django	1.5.12	https://www.djangoproject.com/	Django is a high-level Python Web framework used for web development
django-smart-load-tag	0.3.2 (Dated: 1/9/12)	https://pypi.python.org/pypi/django-smart-load-tag/0.3.2	django-smart-load-tag brings namespaces and more control to Django's {% load %} tag
document-register-element (Polyfills)	0.5.3	https://github.com/WebReflection/document-register-element	A library that normalizes browser API for creating custom HTML elements.
domelementtype	1.1.3	https://github.com/FB55/domelementtype	Provides all the types of nodes in htmlparser2's dom
domhandler	2.2.1	https://github.com/fb55/DomHandler	The DOM handler (formally known as DefaultHandler) creates a tree containing all nodes of a page. The tree may be manipulated using the domutils library.
doT	0.1.6	https://github.com/olado/doT	Javascript template engine for nodejs and browsers
duo_python	9c3157324bb6125246282ddef0cc4560bc3dc5c4	https://github.com/duosecurity/duo_python/	We only use a javascript file from this to integrate with Duo's Two-Factor-Authentication product. We don't use the python pieces themselves, nor do we use the "appengine-openid-provider" or "cookie.py" components that are under a different license.
entities	1.0.0	https://github.com/fb55/entities	En- & decoder for XML/HTML entities.
Envoy	0.0.2 (Dated: 10/15/11)	https://pypi.python.org/pypi/envoy/	Envoy is an API for running external processes
Firebug Lite	1.0 or 1.1	https://getfirebug.com	Firebug integrates with Firefox to put web development tools at your fingertips while you browse. You can edit, debug, and monitor CSS, HTML, and JavaScript live in any web page
FormEncode	0.4	http://www.formencode.org	FormEncode is a validation and form generation package.
handlebars.js	1.0.beta.2	http://handlebarsjs.com/	Handlebars.js is an extension to the Mustache templating language created by Chris Wanstrath. Handlebars.js and Mustache are both logicless templating languages that keep the view and the code separated like we all know they should be.
history.js	1.7.1 (Dated: 10/4/11)	https://github.com/browserstate/history.js	History.js supports the HTML5 History/State APIs (pushState, replaceState, onPopState) in all browsers
htmlparser2	3.8.0	http://github.com/fb55/htmlparser2	A forgiving HTML/XML/RSS parser. The parser can handle streams and provides a callback interface.
HTTP Client	4.3.6	http://hc.apache.org/httpclient-legacy/index.html	Although the java.net package provides basic functionality for accessing resources via HTTP, it doesn't provide the full flexibility or functionality needed by many applications. The Jakarta Commons HttpClient component seeks to fill this void by providing an efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations.
HTTP Core	4.3.3	http://hc.apache.org/httpcomponents-core-ga/	HttpCore is a set of low level HTTP transport components that can be used to build custom client and server side HTTP services with a minimal footprint. HttpCore supports two I/O

			models: blocking I/O model based on the classic Java I/O and non-blocking, event driven I/O model based on Java NIO.
httplib2	0.9.2	https://github.com/jcgregorio/httplib2/	httplib2 is a HTTP client library which supports many features left out of other HTTP libraries.
ICU	33817	http://site.icu-project.org	ICU is a widely used set of C/C++ and Java libraries providing Unicode and Globalization support for software applications. ICU converts various code pages to Unicode.
intro.js	1.0.0	https://github.com/usablica/intro.js	Better introductions for websites and features with a step-by-step guide for your projects.
IPy	0.70 (Dated: 11/23/10)	http://c0re.23.nu/c0de/lPy	IPy provides a Python class and tools for handling of IPv4 and IPv6 addresses and networks.
Jackson Annotations	2.5.3	https://github.com/FasterXML/jackson-annotations	Core annotations (annotations that only depend on jackson-core) for Jackson data processor
Jackson Core	2.5.3	https://github.com/FasterXML/jackson-core	This project contains core low-level incremental ("streaming") parser and generator abstractions used by Jackson Data Processor. It also includes the default implementation of handler types (parser, generator) that handle JSON format.
Jackson Databind	2.5.3	https://github.com/FasterXML/jackson-databind/	This project contains the general-purpose data-binding functionality and tree-model for Jackson Data Processor. It builds on core streaming parser/generator package, and uses Jackson Annotations for configuration.
jemalloc	4.0.4	http://www.canonware.com/jemalloc/	jemalloc is a general purpose malloc(3) implementation that emphasizes fragmentation avoidance and scalable concurrency support
jg_global	0.9	Jason Gatt	Global functions for defining classes and managing resources
jg_lib	unversioned	Not available online	A set of building blocks JavaScript for building complex UIs and data visualizations.
Joda Time	2.8.1	http://www.joda.org/joda-time/	Joda-Time provides a quality replacement for the Java date and time classes. The design allows for multiple calendar systems, while still providing a simple API. The 'default' calendar is the ISO8601 standard which is used by XML. The Gregorian, Julian, Buddhist, Coptic, Ethiopic and Islamic systems are also included, and we welcome further additions. Supporting classes include time zone, duration, format and parsing.
jqTree	0.19	https://mbraak.github.io/jqTree/	jqTree is a tree widget for jQuery
jQuery	1.8.2, 2.1.0	http://jquery.com/	jQuery is a JavaScript library that makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.
jQuery bgiframe	2.1.2	https://github.com/brandonaaron/bgiframe	jQuery bgiframe is a jQuery plugin that helps deal with IE z-index issues.
jQuery deparam	0.1.0	https://github.com/AceMetrix/jquery-deparam	Extracted $.deparam from Ben Alman's jquery-bbq with license info included. Deparam is the inverse of jquery's $.param method. It takes a parameterized querystring and converts it back into an object. The format is in many ways a more compact way to serialize a javascript object over JSON. For example (from the included tests):
jQuery dimensions	1.2	http://api.jquery.com/category/dimensions/	The jQuery dimensions methods are used to get and set the CSS dimensions for the various properties.
jQuery File Upload	5.40.1	https://github.com/blueimp/jQuery-File-Upload	jQuery File Upload is a widget with multiple file selection, drag and drop support, progress bars, validation and preview images, audio and video for jQuery.
jQuery Form	2.8.4	http://malsup.com/jquery/form/	The jQuery Form Plugin allows you to easily and unobtrusively upgrade HTML forms to use AJAX.
jQuery Iframe Transport	1.8.2	https://github.com/blueimp/jQuery-File-Upload	jQuery IFrame Transport is a plugin for jQuery File Upload
jQuery nestedSortable	1.1.1	http://mjsarfatti.com/sandbox/nestedSortable	jQuery nested sortable is a jQuery plugin that extends Sortable UI functionalities to nested lists
jQuery QUnit	1.14.0	https://github.com/jquery/qunit	jQuery quint is a JavaScript Unit Testing framework
jQuery Sparkline	2.1.3	http://omnipotent.net/jquery.sparkline	This jQuery plugin generates sparklines (small inline charts) directly in the browser using data supplied either inline in the HTML, or via javascript
jQuery tablesorter	2.0.3 (Dated: 3/17/08)	http://tablesorter.com	jQuery tablesorter is a jQuery plugin that does client-side table sorting
jQuery TipTip	1.3 (Dated 3/23/10)	code.drewwilson.com/entry/tiptip-jquery-plugin	jQuery TipTip is a jQuery plugin that will create a custom tooltip to replace the default browser tooltip

JQuery Tools	1.2.5	http://jquerytools.github.io	jQuery Tools is a collection of user interface components for websites
jquery trap input	1.1.0	https://github.com/julienw/jquery-trap-input	The jQuery trap input plugin implements input trapping as described by the Web Accessibility Initiative.
jQuery Treeview	1.4.1 (Dated: 10/22/10)	http://bassistance.de/jquery-plugins/jquery-plugin-treeview/	jQuery Treeview is jQuery plugin, that provides interactive trees.
jQuery UI	1.8.5, 1.9.0, 1.10.4	http://jqueryui.com	jQuery UI is a set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.
jQuery UI selectmenu	1.1.0	http://jqueryui.com/selectmenu/	The jQuery selectmen component duplicates and extends the functionality of a native HTML select element to overcome the limitations of the native control.
jQuery UI Spinner	1.10.4	http://jqueryui.com/about	jQuery UI Spinner allows users to increment or decrement text box values without having to input it manually
jquery.resize	1.1	https://github.com/cowboy/jquery-resize	A plugin for the jquery library for reacting to changes in UI component sizes
jquery-cookie	r548 (Dated:11/8/06)	https://code.google.com/p/jqueryjs/source/diff?spec=svn548&r=548&format=side&path=/trunk/plugins/cookie/cookie.js	Used to create a cookie
jquery-placeholder	1.7	https://github.com/mathiasbynens/jquery-placeholder	A jQuery plugin that enables HTML5 placeholder behavior for browsers
jsdom	1.1.0	https://github.com/tmpvar/jsdom	JavaScript implementation of the WHATWG DOM and HTML standards, for use with io.js
jsep	0.3.0-beta	http://jsep.from.so	Used to parse eval-expression syntax.
jsmin	1.0 (by date: 1/9/12)	http://www.crockford.com/javascript/jsmin.html	JSMin is a .C file that provides functionality which removes comments and unnecessary whitespace from JavaScript files.
JSON	2014-02-04	https://github.com/douglascrockford/JSON-js	JSON (JavaScript Object Notation) is a lightweight data-interchange format based on a subset of the JavaScript Programming Language
Leaflet	0.4.5 (Dated: 10/25/12)	http://leafletjs.com	Leaflet is an open-source JavaScript library for mobile-friendly interactive maps
less.js	2.3.0	http://lesscss.org/	Less is a CSS pre-processor, meaning that it extends the CSS language, adding features that allow variables, mixins, functions and many other techniques that allow you to make CSS that is more maintainable, themable and extendable.
Libarchive	3.1.2 (Dated: 2/9/13)	http://www.libarchive.org	Libarchive is a multi-format archive and compression library
libffi	3.0.9	https://github.com/atgreen/libffi	A portable foreign-function interface library
Libxml2	2.9.3	http://xmlsoft.org	Libxml2 is the XML C parser and toolkit
Libxslt	1.1.28 (Dated: 11/21/12)	http://xmlsoft.org	Libxslt is the XSLT C library based on libxml2
LogCabin	None	http://starship.python.net/~skippy/win32	Provides consensus based captain election protocol
LowPro for jQuery	None	https://github.com/danwrong/low-pro-for-jquery	Low Pro is a A jQuery port of the Low Pro behavior framework that was originally written for Prototype
lxml	3.3.5	http://lxml.de	lxml is a library for processing XML and HTML in the Python language
lz4	r128	https://code.google.com/p/lz4/	LZ4 is a very fast lossless compression algorithm with near-linear scalability for multi-threaded applications. It also features an extremely fast decoder
M2Crypto	0.21.1	https://github.com/martinpaljak/M2Crypto	M2Crypto is a Python wrapper for OpenSSL
Mako	1.0.1	http://www.makotemplates.org	Mako is a template library written in Python that provides a familiar, non-XML syntax which compiles into Python modules
Moment.js	2.8.3	http://momentjs.com/	moment.js parses, validates, manipulates, and displays dates in JavaScript.
mongo-c-driver	1.1.8	https://github.com/mongodb/mongo-c-driver	This is the C Driver we use to interface with MongoDB.
nan	1.3.0	https://github.com/nodejs/nan	A header file filled with macro and utility goodness for making add-on development for Node.js easier across versions 0.8, 0.10 and 0.12 as well as io.js.
node	0.12.7	https://nodejs.org/	Node is a platform built on Chrome's JavaScript runtime for building fast, scalable network applications
node-dispatch	Unknown	https://github.com/caolan/dispatch	A really simple URL dispatcher for Connect or a plain Node.js HTTP Server. Allows arbitrarily nested regular expressions for matching URLs and calling an associated

			function.
node-http-proxy	0.8.1 (Dated: 6/5/12)	https://github.com/nodejitsu/node-http-proxy	node-http-proxy is a http proxy for node.js
numeral.js	1.5.3	http://numeraljs.com/	numeral.js is a javascript library for formatting and manipulating numbers
NVD3	1.0.0b	http://nvd3.org/	NVD3 provides support to create re-usable charts for d3.js
nwmatcher	1.3.3	http://javascript.nwbox.com/NWMatcher/	A fast CSS selector engine and matcher.
OpenLDAP	2.4.43	http://www.openldap.org	Authentication library used for authentication with LDAP sources
OpenSSL	1.0.2e	http://www.openssl.org	The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS) protocols as well as a full-strength general purpose cryptography library.
parse5	1.1.6	http://inikulin.github.io/parse5/	WHATWG HTML5 specification-compliant, fast and ready for production HTML parsing/serialization toolset for Node and io.js.
Pcre	8.38	http://www.pcre.org	The PCRE library is a set of functions that implement regular expression pattern matching using the same syntax and semantics as Perl 5.
pdfkit	0.1.5	http://pdfkit.org/	PDFKit is a PDF document generation library for Node and the browser
ptmalloc2	(2006-05-31)	http://www.malloc.de/en/	Provides fast dynamic memory allocators for multithreaded systems around and also causes remarkably low memory
pyOpenSSL	0.13.1	https://github.com/pyca/pyopenssl	Provides access to openssl from python. It is shipped as a library and called by the python processes (it's dynamically linked and imported by python dynamically)
PyPNG	0.0.13	https://github.com/drj11/pypng/	PyPNG is a Python library for PNG image encoding/decoding
Python	2.7.11	http://www.python.org	Python is an interpreted, object-oriented, high-level programming language
Python - uuid	1.3.0	http://zesty.ca/python/uuid.html	UUID object and generation functions
pywin32	217	http://starship.python.net/~skippy/win32	Python for Windows Extensions
reportlab	2.6 (Dated: 9/27/12)	http://www.reportlab.com/	ReportLab PDF Toolkit allows rapid creation of rich PDF documents, and also creation of charts in a variety of bitmap and vector formats.
Requests	2.3.0 (Dated: 5/116/14)	https://github.com/kennethreitz/requests	Requests is an HTTP library written in Python
require-css	0.1.1 (Dated 12/16/13)	https://github.com/guybedford/require-css	require-css is a requireJS CSS loader plugin to allow CSS requires and optimization
RequireJS	2.1.15	http://github.com/jrburke/requirejs	RequireJS is a file and module loader for JavaScript
sax js	1.1.4	https://github.com/isaacs/sax-js	A sax-style parser for XML and HTML. Designed with node in mind, but should work fine in the browser or other CommonJS implementations.
script.js	© 2011	http://www.dustindiaz.com/scriptjs	script.js is an asynchronous JavaScript loader and dependency manager
Select2	3.4.6	https://select2.github.io/	Select2 gives you a customizable select box with support for searching, tagging, remote data sets, infinite scrolling, and many other highly used options.
Snappy	1.1.1.7	https://github.com/xerial/snappy-java	The snappy-java is a Java port of the snappy http://code.google.com/p/snappy/, a fast C++ compresser/decompresser developed by Google.
spin.js	1.2.7	http://fgnass.github.io/spin.js/	Spin.js dynamically creates spinning activity indicators that can be used as resolution-independent replacement for AJAX loading GIFs.
sprintf.js	810.1015	http://www.diveintojavascript.com/projects/javascript-sprintf	sprintf for JavaScript is a complete open source JavaScript sprintf implementation
SQLite	3.8.11.1	www.sqlite.org	SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine
strftime.js	0.1	http://dren.ch/strftime/	strftime.js formats date and time strings
SWFObject	2.2 beta1 (Dated: 6/3/09)	https://github.com/swfobject/swfobject	SWFObject is a method to embed Flash content
text.js	2.0.13	http://github.com/requirejs/text	text.js is an AMD loader plugin for loading text resources
tzcode	2013c	http://www.iana.org/time-zones	The Time Zone Database contains code and data that represent the history of local time for many representative

			locations around the globe
tzdata	2013c	http://www.iana.org/time-zones	The Time Zone Database contains code and data that represent the history of local time for many representative locations around the globe
ubuntu-font-family	0.80	http://font.ubuntu.com/	This is the Ubuntu Font Family
UCD (Unicode character database)	8.0.0	http://www.unicode.org/Public/UNIDATA	The Unicode Character Database (UCD) consists of a number of data files listing Unicode character properties and related data.
Underscore.js	1.6.0	http://underscorejs.org/	Underscore is a JavaScript library that provides over 100 functions that support both functional helpers like map, filter, invoke u2014 as well as more specialized functionality like function binding, javascript templating, creating quick indexes, deep equality testing, and so on.
xmlhttprequest	1.6.0	https://github.com/driverdan/node-XMLHttpRequest	node-XMLHttpRequest is a wrapper for the built-in http client to emulate the browser XMLHttpRequest object.
xml-name-validator	2.0.1	https://github.com/jsdom/xml-name-validator	Validates whether a string matches the production for an XML name or qualified name. This is a child dependency for jsdom which is used in our pdf print server to render JavaScript and rasterize it as a PDF.
xmlsec1	1.2.20	https://www.aleksey.com/xmlsec	XML Security Library
XRegExp	2.0.0	http://xregexp.com	XRegExp is a JavaScript library that provides augmented and extensible regular expressions. You get new syntax, flags, and methods beyond what browsers support natively.
xxhash	r39	https://github.com/Cyan4973/xxHash	xxhash is an extremely fast non-cryptographic hash algorithm
zlib	1.2.8	http://zlib.net/zlib.html	zlib is designed to be a lossless data-compression library for use on virtually any computer hardware and operating system.

Table 8-3: TOE Libraries

8.7.4 FPT_TUD_EXT.1:

The TOE provides the ability for administrators to determine its currently installed version by using the Help→About in the GUI or through the underlying platform's package manager.

Splunk automatically checks to see if an update is available. Splunk will notify the users via the login screen that there is an update available. Splunk does not download the update 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. RPM can also be used to show the current version of the TOE. Splunk provides a public key that is installed to RPM in the evaluated configuration. An administrator can then run "rpm -K" in order to verify the update against the installed public key prior to installation.

All python files used by the TOE are .py files and not .pyc so the TOE itself will not replace its own modify or replace its own executable code. Code updates can only be performed using the platform's package manager.

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.

8.8 Trusted Path/Channels

8.8.1 FTP_DIT_EXT.1:

The TOE uses HTTPS / TLSv1.2 to secure data in transit over trusted channels and paths. The TOE acts

as an HTTPS server for remote administration performed using the Web GUI. The TOE also has the ability to transmit alert notifications to an SMTP server in the Operational Environment so that administrators can receive email notifications of certain events. If the SMTP server is not located locally to the TOE, this communications channel can also be secured using HTTPS. Once the messages reach the SMTP server, they have been transmitted to the Operational Environment so any subsequent security of the email transmissions themselves (such as S/MIME) cannot be enforced using the TSF. The TOE includes a single cryptographic implementation that is used to handle both the SMTP trusted channel and the administrative trusted path. For intra-TOE communications where the Splunk instance on one system is configured as a forwarder to pass system generated data to the primary instance, security is achieved using TLSv1.2 with the primary instance acting as the server.

Appendix A: Platform APIs Supported by the TOE

Splunk Enterprise ships almost all of the libraries and scripting languages Splunk requires to operate and does not depend on the platform. The list of components shipped with Splunk is captured in Table 8-3. So, 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 dynamically linked libraries and system calls (APIs).

Dynamically Linked Libraries provided by RHEL	Packages Providing These Libraries
libcom_err.so.2	libcom_err-1.41.12-18.el6.x86_64
libc.so.6	glibc-2.12-1.192.el6.x86_64
libdl.so.2	glibc-2.12-1.192.el6.x86_64
libgssapi_krb5.so.2	krb5-libs-1.10.3-10.el6_4.6.x86_64
libk5crypto.so.3	krb5-libs-1.10.3-10.el6_4.6.x86_64
libkeyutils.so.1	keyutils-libs-1.4-4.el6.x86_64
libkrb5.so.3	krb5-libs-1.10.3-10.el6_4.6.x86_64
libkrb5support.so.0	krb5-libs-1.10.3-10.el6_4.6.x86_64
libm.so.6	glibc-2.12-1.192.el6.x86_64
libpthread.so.0	glibc-2.12-1.192.el6.x86_64
libresolv.so.2	glibc-2.12-1.192.el6.x86_64
librt.so.1	glibc-2.12-1.192.el6.x86_64
libselinux.so.1	libselinux-2.0.94-7.el6.x86_64
libutil.so.1	glibc-2.12-1.192.el6.x86_64
/lib64/ld-linux-x86-64.so.2	glibc-2.12-1.192.el6.x86_64

Below is the list of system calls (platform APIs):

mmap, inet_ntoa, ftello64, wait4, __errno_location, getc, getrlimit, __memmove_chk, getpwnam_r, wcrtomb, read, pthread_barrier_destroy, isspace, getpwuid_r, fgets, getopt_long, readdir_r, getpgrp, setsid, __stack_chk_fail, __memset_chk, fflush, strlen, gmtime_r, munmap, strtoll, strrchr, sigprocmask, memset, dlopen, __isinf, getdtablesize, fchown, pthread_cond_broadcast, inet_pton, setgroups, poll, rmdir, pwrite, pthread_mutex_destroy, __xpg_strerror_r, strspn, getpeername, memmove, asctime_r, setregid, wait, calloc, accept, pthread_sigmask, pthread_mutex_init, pthread_rwlock_wrlock, dl_iterate_phdr, ungetc, setsockopt, readv, socket, usleep, flockfile, sinh, inet_ntop, getpgid, pthread_mutex_trylock, __isinff, system, isatty, gethostbyname, ftell, puts, backtrace, fsync, dirname, __cxa_atexit, qsort, connect, dup, readdir64, memcpy, lrint, getwc, pwrite64, __strncat_chk, pathconf, wmemmove, fmod, isgraph, lrand48, __strncpy_chk, pthread_create, atoi, asin, pthread_mutex_lock, lseek64, pthread_setspecific, getcwd, tcgetpgrp, setenv, ferror, pthread_once, mprotect, __printf_chk, recvfrom, dladdr, posix_memalign, setreuid, fgetc, mbsrtowcs, tmpfile, lseek, chdir, getnameinfo, pthread_barrier_wait, strstr, setrlimit, sem_timedwait, isprint, tcsetpgrp, malloc, sscanf, sigaltstack, closedir, logf, __lxstat64, __ctype_get_mb_cur_max, setpriority, ftruncate64, getloadavg,

pthread_mutexattr_init, socketpair, __strdup, execve, getsid, setegid, pread64, __strcpy_chk, truncate, mkdir, killpg, statfs, fchdir, tempnam, geteuid, sleep, setvbuf, log10, btowc, fopen64, __memcpy_chk, alphasort64, getgrgid_r, bind, pause, posix_fadvise, setpgrp, sched_yield, mbsnrtowcs, setpgid, sendfile64, inet_aton, strncmp, __stpcpy_chk, funlockfile, ioctl, getservbyname, pthread_rwlock_rdlock, fputc, dlerror, localeconv, __sysv_signal, strncpy, msync, isupper, fstatfs, wcsnrtombs, inet_addr, strcpy, getpriority, islower, memrchr, getpagesize, sprintf, pthread_rwlock_tryrdlock, unsetenv, pthread_cond_init, nice, clock_getres, getitimer, sigaction, frexp, pthread_attr_setstacksize, getegid, getaddrinfo, cos, execv, fcntl, getpwnam, sqrtf, sysconf, open, exp, pthread_cond_wait, gethostbyaddr, getppid, getgroups, getlogin, memcmp, __xmknod, strncasecmp, confstr, pthread_mutexattr_settype, wmemcpy, strchr, sigaddset, freeaddrinfo, dup2, hypot, pthread_attr_init, cfmakeraw, lchown, sem_destroy, atan, waitpid, rename, fprintf, __ctype_toupper_loc, writev, pow, ceil, perror, ldexp, ceilf, pthread_condattr_destroy, fork, pthread_barrier_init, select, ungetwc, srand, __fxstat, _exit, pthread_kill, getsockname, sem_init, iscntrl, gethostname, fesetround, shutdown, forkpty, unlink, __read_chk, clock_gettime, ttyname, tzset, pthread_rwlock_unlock, popen, strftime, utime, putenv, strerror, uname, ctermid, __h_errno_location, umask, endgrent, mbrtowc, setrlimit64, feof, __fxstat64, raise, nl_langinfo, wmemchr, listen, __rawmemchr, sendto, fchmod, strtol, wcscmp, statvfs, __realpath_chk, backtrace_symbols, putc, fpathconf, atan2, remove, clock, sqrt, getrlimit64, bindtextdomain, mkstemp, __strcat_chk, sendmsg, __pread64_chk, pthread_key_create, fputs, nanosleep, setitimer, chroot, getenv, __fprintf_chk, tmpfile64, recvmsg, floor, sysinfo, __libc_current_sigrtmin, ftruncate, readlink, fread, __finite, link, setgid, recv, fdatasync, pthread_attr_setscope, times, rand, getrusage, getgid, sync, mktime, textdomain, nftw, pthread_condattr_init, setuid, __isnan, tanh, putchar, sigemptyset, setpwent, getpid, __libc_current_sigrtmax, realloc, fscanf, exit, getpwuid, pread, pthread_mutex_unlock, open64, fileno, __lxstat, isxdigit, pclose, pthread_cond_destroy, scandir64, round, isalnum, fdopen, setlocale, mkdtemp, strtoull, cosh, srand48, fstatvfs64, vsnprintf, pthread_getspecific, strncat, __libc_start_main, fseeko64, localtime, __vsnprintf_chk, sem_trywait, time, pthread_rwlock_destroy, __xstat, utimes, mmap64, readdir, pipe, pthread_cond_timedwait, dlsym, strsignal, pthread_rwlock_init, strtok, floorf, isalpha, __ctype_b_loc, snprintf, initgroups, wmemset, tan, getuid, fwrite, clearerr, fopen, chown, close, if_nametoindex, pthread_mutexattr_destroy, kill, ispunct, ctime, getpwent, pthread_cond_signal, symlink, wctob, pthread_detach, endpwent, pthread_rwlock_trywrlock, localtime_r, mkfifo, tcsetattr, strcat, sem_post, sin, openpty, wmemcmp, __vfprintf_chk, fseeko, strcmp, pthread_attr_destroy, pthread_join, wcslen, strtoul, __sprintf_chk, getgrent, sincos, execvp, getgrnam_r, gettext, statvfs64, tcgetattr, getsockopt, sigfillset, stpcpy, fseek, __ctype_tolower_loc, dlclose, write, difftime, gettimeofday, log, opendir, strtod, __snprintf_chk, __xstat64, strcasecmp, strcspn, fclose, memchr, sem_wait, free, acos, __assert_fail, alarm, siginterrupt, rewind, pthread_self, modf, abort, putwc, seteuid, access, pthread_key_delete, ftello, send, chmod, gai_strerror, signal, mkstemp64, __strtof_l, __wcsxfrm_l, __strxfrm_l, __wctype_l, __towlower_l, __iswctype_l, stdout, setresuid, freeifaddrs, getifaddrs, __fdelt_chk, __strcoll_l, prctl, strtold_l, __wcsftime_l, qsort_r, getresgid, getresuid, __strndup, __pthread_key_create, __uselocale, __open64_2, __strtod_l, __nl_langinfo_l, __freelocale, __fread_chk, stdin, timegm, __duplocale, __wcscoll_l, preadv64, stderr, __strftime_l, tmpnam_r, syscall, wait3, __tls_get_addr, setlinebuf, __towupper_l, timespec_get, pthread_setname_np, __newlocale, setresgid, __uflow, pwritev64, __isoc99_sscanf, __isoc99_fscanf,