Splunk Inc. Splunk 4.1.7 Security Target

Version 2.0
February 1, 2011

Prepared for:
Splunk Inc.
250 Brannan Street, 2nd Floor,
San Francisco, CA 94107

Prepared by:
Booz Allen Hamilton
Common Criteria Testing Laboratory
900 Elkridge Landing Road, Suite 100
Linthicum, MD 21090-2950

Table of Contents

List of Figures

List of Tables

1 Security Target Introduction

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

1.1 ST Reference

This section provides information needed to identify and control this ST and its Target of Evaluation. This ST targets Evaluation Assurance Level 2 (EAL2) augmented with ALC_FLR.1.

1.1.1 ST Identification

ST Title:	Splunk Inc. Splunk 4.1.7 Security Target
ST Version:	2.0
ST Publication Date:	February 1, 2011
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this ST provides identifying information for the TOE. It includes an ST Introduction, ST Reference, ST Identification, TOE Reference, TOE Overview, and TOE Type.

Chapter 2 describes the TOE Description, which includes the physical and logical boundaries, and describes the components and/or applications that are excluded from the evaluated configuration.

Chapter 3 describes the conformance claims made by this ST.

Chapter 4 describes the Security Problem Definition as it relates to threats, Operational Security Policies, and Assumptions met by the TOE.

Chapter 5 identifies the Security Objectives of the TOE and of the Operational Environment.

Chapter 6 describes the Extended Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs).

Chapter 7 describes the Security Functional Requirements.

Chapter 8 describes the Security Assurance Requirements.

Chapter 9 is the TOE Summary Specification (TSS), a description of the functions provided by the TOE to satisfy the SFRs and SARs.

Chapter 10 is the Security Problem Definition Rationale and provides a rationale or pointers to a rationale, for security objectives, assumptions, threats, requirements, dependencies, and PP claims for the chosen EAL, any deviations from CC Part 2 concerning SFR dependencies, and a mapping of threats to assumptions, objectives, and SFRs. It also identifies the items used to satisfy the Security Assurance Requirements for the evaluation.

1.1.3 Terminology

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

Terminology	Definition
Access Control Lists	The owner of each created object specifies the read/write access available to each role within the system. Obviously, the owner of an object has unrestricted access to the objects it controls.
Authorized User	A user that has been assigned a role with the attributes that allow an action on an object as defined in Table 7-3. This essentially means "any user that is capable of performing the action in question."
Capability	An action in the TOE that can be added to a role to grant the role the ability to perform said action.
Deployment Client	This refers to all Splunkd processes that are sent configuration updates by the Deployment Server.
Deployment Server	An instance of the Splunkd process that is configured solely to deploy configuration updates to the other Splunkd processes within the TOE deployment.
Forwarder	An instance of the Splunkd process that is configured solely to collect raw IT data logs, parse them, formulate indexed logs, and then forward both the raw data and the indexed logs to another configured Splunkd process.
Index	When used as a verb, this refers to the actual process of parsing raw data logs, extracting fields, and storing the parsed data. When used as a noun, this refers to where said parsed data is stored upon Splunkd processes configured as indexers.
Indexed Data	This refers to parsed IT data that is stored in an indexer.
Indexer	An instance of the Splunkd process that is configured to collect parsed data logs as well as raw data logs from a forwarder and to store said data.
IT Data	All data that the TOE collects and indexes.
Parsing	Specifically to Splunk, this is the act of utilizing Splunk's indexing functionality to process raw data and extract specific default and user-defined fields. The output of this process is indexed data.
Raw Data	Unprocessed IT data the TOE collects from any configured source.

Deleted: u

Deleted: d

Deleted: d

Receiver	Any Splunkd process that receives data from one or more forwarders.
Role	A named bundle of capabilities and allowed indexes that determines the amount of access specific users are allowed within the TOE. There are defaults, but additional roles can be user-generated. Roles are assigned to users.
Search-head	An instance of the Splunkd process that is configured solely to be the primary component for searching. It is also the only Splunkd process within the evaluated configuration to interface with users via the Splunk Web and Splunk CLI processes. This means that most of the general TOE management is utilized through this process exclusively.
Server Class	A deployment configuration shared by a group of deployment clients. A deployment client can belong to multiple server classes.
Splunk Object	A Splunk object is an object within the system that has an ACL defined for it.

Table 1-1: Customer Specific Terminology

Term	Definition
Authorized User	A user who, in accordance with proper authentication/authorization, is allowed to perform an operation.
External IT Entity	Any IT product or system, trusted or not, outside of the TOE that interacts with the TOE.
TOE Security Functions (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Table 1-2: CC Specific Terminology

1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 1-3. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
AD	Active Directory
ACL	Access Control List
CA	Certificate Authority
CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Scheme
CCIMB	Common Criteria Interpretations Management Board

CLI	Command-line Interface
CPU	Central Processing Unit
EAL	Evaluation Assurance Level
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
NIAP	National Information Assurance Partnership
OS	Operating System
PP	Protection Profile
RAID	Redundant Array of Independent Disks
RBAC	Role Based Access Control
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
ST	Security Target
TCP	Transmission Control Protocol
TOE	Target of Evaluation
TSF	TOE Security Function
UDP	User Datagram Protocol
UI	User Interface
URL	Uniform Resource Locator
X.509	X.500 Series Standard for Public-key and Attribute and Certificate Frameworks

Table 1-3: Acronym Definitions

1.1.5 References

[1] Common Criteria for Information Technology Security Evaluation, CCMB-2009-07-004, Version 3.1 Revision 3, July 2009

[2] Splunk Admin Manual Version 4.1.7

[3] Splunk User Manual Version 4.1.7

[4] Splunk Installation Manual Version 4.1.7

[5] Splunk Knowledge Manager Manual Version 4.1.7

[6] Splunk Developer Manual Version 4.1.7

[7] Splunk Search Reference Version 4.1.7

[8] Splunk Release Notes Version 4.1.7

[9] Splunk Application Management Version 4.1.7

1.1.6 CC Concepts

The following are CC concepts as used in this document. A Subject is any user of the TOE (account, user, administrative user). An Object (i.e., resource or entity) can be a dataset, volume, command issued by a user, etc. An Operation is any action on a resource (e.g. read, write, create, fetch, update, control, alter, or scratch). A Security Attribute is information such as username, groups, profiles, facilities, passwords, etc. that is kept in the security file for the user. An External Entity is anything outside of the TOE that affects the TOE.

1.2 TOE Reference

Splunk 4.1.7

1.3 TOE Overview

Splunk 4.1.7, (herein referred to as Splunk or the TOE), collects IT data logs from various configured machines, stores the logs on disk, and indexes the data it collects. Splunk features broad search functionality to query these logs at based on user requests. Multiple instances of the Splunk process can be utilized in synchronization to optimize the functionality, with different Splunk processes focusing on collecting and forwarding IT data, storing and indexing IT data, and searching IT data and providing a collaborative user interface.

The TOE:

- Collects IT data logs from configured machines
- Stores and indexes collected IT data logs in indexers
- Allows users to perform comprehensive search actions to query IT data logs

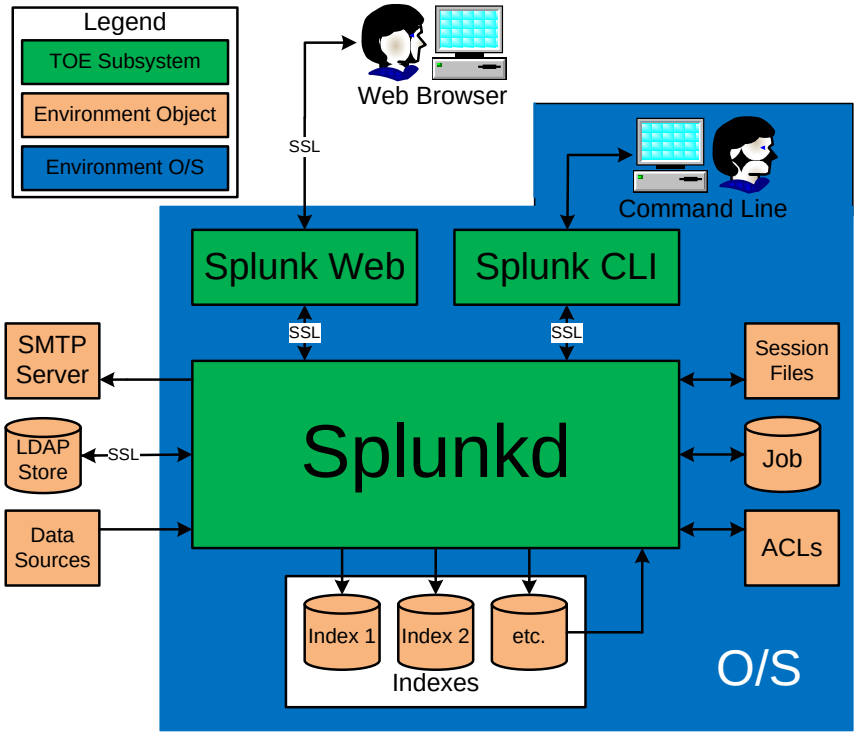

Figure 1 – Splunk TOE Boundary

As illustrated in Figure 1, the TOE boundary contains 3 subsystems: Splunk Web, Splunk, and Splunkd. Each TOE component is a distinct process running on the host machine. Splunk Web and Splunk are accessed through a supported web browser or command-line interface, respectively. Splunkd is the process that provides most of the TOE functionality. In addition to the TOE subsystems and user interfaces, the TOE receives data from configured data sources, be it from the local machine or other machines on the network. It can also connect to an LDAP Store for LDAP authentication. The indexes and search job results are stored on the local filesystem and provide indexing and searching functionality. The TOE can also connect with an SMTP server to send out configured alerts. A more detailed description of each of the TOE subsystems is provided in the paragraphs below.

Splunk Web is a Browser-based UI that supports Firefox 3.5 and Internet Explorer 7 or higher. It is one of the primary ways that a user can interact with the Splunkd subsystem, displaying a subset of Splunk user functionality in the form of commands and fields. Users are challenged to login using username and password. However, no actual parsing or authentication is done using this subsystem; it just prepares and sends commands to

the Splunkd subsystem. All identification, authentication, and authorization functions, as well as performing user actions or queries are sent to the Splunkd subsystem for processing. Users can also use this subsystem and this subsystem only to create charts and views for display.

Splunk CLI is the subsystem that consists of the Command-Line Interface. It has the same functionality as the Splunk Web subsystem except for visual presentation functionality, such as dashboards, charts, graphs, and typeahead. A user uses this subsystem by navigating the operating system's standard command-line interface to the folder in which the "splunk" process resides. The user then issues the command "splunk" to run the executable, but also adds the action the user wishes to perform as command-line arguments. For instance, a user would enter "splunk stop" to stop the Splunkd process.

Splunkd is the subsystem that consists of most of the functionality in the TOE. This subsystem handles identification, authentication, and authorization to interact with or enact actions, objects, or indexes. The primary functionality of the TOE 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.

On the Splunkd side, the primary functionality of the TOE is to gather data from data sources such as Windows, web servers, firewalls, or other IT products. In this regard, each Splunkd instance can be configured to function in its own manner with collecting and parsing IT data. There are several common Splunkd configurations. The Splunkd configurations within the TOE include the following:

- Search-heads, which are the Splunkd processes that the Splunk CLI and Splunk Web subsystems, and by extension the user, can connect to and interact with.
- Forwarders which take in data from data sources and are configured to process and forward that data to other Splunkd processes.
- Indexers which take in data from data sources and/or forwarders, process it, and store it in indexes on the local machine. They also receive and process search requests.
- Deployment servers that answer configuration requests from other Splunkd processes and deploy configuration updates

In addition, once a search request is made using a search-head, a new Splunkd process is started up to process the search. This Splunkd process is only querying indexes, and is not performing any forwarding or indexing functionality. These search processes end once the search is completed; the rest of the processes are more permanent.

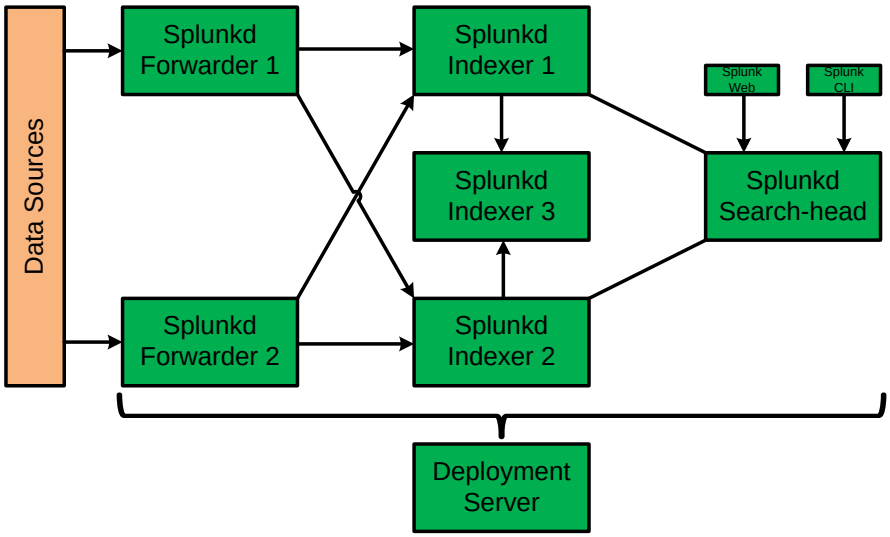

Figure 2 – TOE Deployment

As illustrated in Figure 2, the TOE deployment contains seven instances of Splunkd: two forwarders, three indexers, a deployment server, and a search-head. Each of the Splunkd processes has the same internals, but their functionality changes based on configuration. The reasoning for this deployment is to show the minimum configuration that encapsulates the extent of the TOE functionality. Two forwarders exist, which send data to two of the indexers. These two indexers exist to demonstrate the search-head functionality. The third indexer exists to show the data cloning functionality for high availability functionality. The deployment server exists solely to deploy configuration changes and/or software updates all of the other Splunkd processes, and the search-head is the primary user-facing Splunkd process that coordinates the searching functionality.

1.4 TOE Type

The TOE type for Splunk is Security Management. Security Management is defined by CCEVS as "a set of pervasive security mechanisms which support the security services by direct and supervisory administration, automated processes, and by the activities of all information users."

2 TOE Description

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

2.1 Evaluated Components of the TOE

The following table describes the TOE components in the evaluated configuration:

Component	Definition
Splunk Web	Python and browser-based interface that uses OpenSSL functionality and allows users to perform all management and searching functions on the TOE deployment. This subsystem is only utilized with the Splunkd process designated as the search-head.
Splunk CLI	Command-line interface that uses OpenSSL functionality and allows users to perform most management and searching functions on the TOE deployment. This subsystem is only utilized with the Splunkd process designated as the search-head.
Splunkd	Process that contains most of the Splunk primary functionality. This process contains modules to receive data, forward data, and index data. In addition, when searches are performed, this process performs the search on its configured indexes and returns the results to the search-head. In addition, scheduled searches can be configured by users. This process also contains a web server, handles all authentication and authorization needs, and can be configured to be a deployment server for configuration updates.

Table 2-1: Evaluated Components of the TOE

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
Data Sources	The data sources consist of all products and mechanisms that are configured to send IT data logs to the TOE. This data can include Windows event logs, TCP and UDP syslog events, Windows registry changes, local files, generic scripted, and local Active Directory events.
Indexes	The indexes are the reference pointers that enable fast searching of the parsed logs stored on disk of an indexer. More than one index can exist on an indexer, allowing more advanced TOE users to organize data accordingly.
Job	The job exists on the local storage of the machine of a Splunkd process performing a search. This data store contains the search results of that specific search and will return that data to the search-head.
LDAP Store	This component allows for LDAP authentication instead of normal Splunk authentication. Splunk deployments coordinate with an LDAP store to provide and verify all user attributes.
SMTP Server	This component allows Splunk to send alert emails to configured recipients. These alerts are generally from scheduled searches.

Table 2-2: Evaluated Components of the Operational Environment

2.3 Excluded from the TOE

The following optional products, components, and/or applications can be integrated with Splunk but are not included in the evaluated configuration. They provide no added security related functionality. 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

- PAM, RADIUS, Single sign-on authentication – These authentication mechanisms are not utilized in the evaluated configuration because they are third party and LDAP authentication is sufficient to allow enterprise users in this evaluation.

2.3.2 Installed but Requires a Separate License

No components are installed but require a separate license.

2.3.3 Installed But Not Part of the TSF

These capabilities exist within Splunk, but are not included in the TSF.

- Apps Functionality – This capability of the product allows users to create specific views in the Splunk user interfaces that consist of Splunk objects, which consist of Splunk search data. This capability of the Splunk product is not part of the TSF not only because it does not provide any additional security functions or methods that affect the security functionality of the product, but also because the Splunk App functionality provides for an infinite number of Apps able to be created. Note that the following Apps are included within a standard Splunk installation and are deemed part of the evaluated configuration: Search and Manager.
- Proxy-based Authentication, Scripted Authentication – These authentication mechanisms are not utilized in the evaluated configuration because they are third party and LDAP authentication is sufficient to allow enterprise users in this evaluation.
- "Heavy-weight Forwarders" – This specific forwarder configuration allows for more processing on the forwarder side, but is not required for normal operation of Splunk or the distribution of data between Splunk indexers.

Deleted: a

Deleted: s

Deleted: a

Deleted: f

2.4 Physical Boundary

2.4.1 Hardware

Splunk is a software TOE and therefore has system requirements for the systems upon which it runs. The recommended hardware for both Windows and non-Windows platforms is as follows:

- CPU: 2x quad-core Xeon, 3 GHz

- Memory: 8 GB
- Hard Drive Configuration: RAID 0 or 1+0
- Hard Drive Space: 2 GB for Splunk, additional for indexes (200 GB recommended)

The minimum hardware is as follows:

Platform	CPU	Memory
Red Hat Linux 5.3, Solaris 10	1x 1.4 GHz CPU	1 GB
Windows 2008 R2, Windows 2003, Windows XP, Windows 7	Pentium 4 or equivalent at 2 Ghz	2 GB

Table 2-3: Minimum Hardware Requirements of the TOE

2.4.2 Software

Splunk is supported on the following platforms:

- Windows 2003 (64-bit, supported but not recommended on 32-bit)
- Windows 2008 R2 (64-bit, supported but not recommended on 32-bit)
- Windows XP (32-bit)
- Windows 7 (32-bit, 64-bit)
- Red Hat Linux 5.3
- Solaris 10

Splunk also supports the usage of the following web browsers:

- Firefox 3.5 and 3.6
- Internet Explorer 7 and 8

Note that when using Linux or Solaris installations that the NFS file system should not be used. NFS is usually a poor choice for Splunk indexing activity, for reasons of performance, resilience, and semantics.

2.5 Logical Boundary

The logical boundary of the TOE is described in the terms of the security functionalities that the TOE provides to the systems that utilize this product for intrusion detection.

The logical boundary of the TOE is broken down into the following security classes: IT Data Indexing, Security Audit, Cryptographic Support, User Data Protection, Identification and Authentication, Security Management, Protection of the TSF, and High Availability. Listed below are the security functions with a listing of the capabilities associated with them:

2.5.1 IT Data Indexing

The TOE is able to collect and index IT data from the following log sources: Windows event logs, UDP and TCP syslog, Active Directory, generic scripted inputs, local disk logs, file system changes, and Windows registry changes. Each IT data event has at least the date/time of the event, source, source type, and host name. Only authorized users are able to read the indexed IT data by performing searches on the TOE. Authorized users are able to use the search functionality to search indexed audit logs based upon the data collected during indexing. All IT data logs indexed are protected from deletion or modification. In the evaluated configuration, the TOE is configured to stop indexing and begin to overwrite the oldest IT data records when storage reaches the configured limit, which is synonymous with the storage being full. The TOE also sends an alarm in the form of a user interface banner. In addition, the most recent stored IT records are maintained if storage runs out.

2.5.2 Security Audit

The TOE collects audit logs on TOE startup and shutdown, user login, and any user action on the system, including editing users and configuration. A timestamp is provided, as well as the user who performed the action (if applicable), the action itself, and a success or failure determination. Only authorized users are able to read this audit information by performing searches. The audit data collected is added to the index and is read in the same manner as IT data. The search functionality in the TOE allows authorized users the ability to read audit data to sort and filter the audit data returned to them based upon date/time of the event, type of event, subject identity, and outcome of the event, along with any other search parameters entered. All audit data logs are protected from deletion or modification. The most recent stored audit records are maintained if storage runs out. In the evaluated configuration, the TOE is configured to stop collecting IT data when storage is full and will also send an alarm in the form of a user interface banner. The oldest audit logs will be deleted to make room for new audit logs. However, audit logs can still be generated in this state.

2.5.3 Cryptographic Support

The TOE utilizes OpenSSL packages to generate cryptographic keys utilizing the RSA algorithm with 1024-bit keys. The TOE will overwrite old keys whenever a new key is generated. All sensitive interfaces are protected utilizing these encryption standards, including the user interface connections, connections between TOE components in the deployment, and the optional LDAP server.

2.5.4 User Data Protection

The TOE utilizes an RBAC Policy which requires roles to be assigned to users to perform anything but the most basic functions of the TOE. Roles are assigned a collection of capabilities which are operations that can be performed on specific objects. Roles are also assigned indexes which allow the searching of specific IT/audit data. Additionally, some

objects created in the TOE also contain Access Control Lists (ACLs) that define which roles have read and/or write privileges.

2.5.5 Identification and Authentication

The TOE provides user accounts that have the following attributes: username, password, and roles. All users must successfully identify and authenticate themselves utilizing their username and password combination before they can make any TSF-related actions. There are two authentication mechanisms utilized in the TOE: Splunk authentication and LDAP authentication. Authorized users are able to select the authentication method used within the configuration options of the TOE. Upon authentication, users are bound to their role and other user attributes within a session object. A user session is terminated if the user is deleted or if all roles have been removed for the user. In addition, sessions will be terminated due to inactivity.

2.5.6 Security Management

The security management of the TOE is controlled by user actions that are authorized by the TOE's RBAC policy which is described in Section 9.1.4. Every function within the system, along with the objects it affects, is controlled by specific capabilities, indexes, or ACLs (Access Control Lists) available to the user performing the action. The security attributes are edited and assigned using this same RBAC policy. The TOE provides restrictive default values for these security attributes. Authorized users are able to specify alternative default values based upon the needs of the TOE deployment. The primary security attributes within the system are roles. The default roles in the system are the following: admin, power, user, and can_delete. Additional roles can be generated by authorized users. One or more roles must be assigned to a user before the user can perform any TSF-related action on the TOE. Using the functions within the TOE, authorized users can revoke security attributes for specific users. If a user has their user account removed or if a user has all roles removed for the user, the user will then have any active session terminated. In addition, sessions are updated when a user's role changes.

2.5.7 Protection of the TSF

The TOE utilizes OpenSSL to prevent unauthorized disclosure of data and detect modification of TSF data sent to the LDAP store. Detected modifications of TSF data will result in the packet being dropped. Additionally, OpenSSL is utilized to protect data being transferred between TOE components. OpenSSL is also used to create a logically distinct trusted path between remote users and the TOE, and will protect the TSF data in transit from unauthorized modification or disclosure. Remote users initiate the trusted path to the TOE. The trusted path is required to be used for user authentication, management actions, and data transfer.

2.5.8 High Availability

The TOE also provides mechanisms for high availability. The TOE will maintain a secure state whenever an indexer fails. The TOE will also ensure that the indexing functionality of the product will still operate if a single indexer fails.

3 Conformance Claims

3.1 CC Version

This ST is compliant with *Common Criteria for Information Technology Security Evaluation*, CCMB-2009-07-004, Version 3.1 Revision 3 July 2009.

3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 2 extended for EAL2 to include all applicable NIAP and International interpretations through 15 June 2010.

3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 3 conformant plus flaw remediation for EAL2 to include all applicable NIAP and International interpretations through 15 June 2010.

3.4 PP Claims

This ST does not claim conformance to any Protection Profile.

3.5 Package Claims

This TOE has a package claim of EAL2.

3.6 Package Name Conformant or Package Name Augmented

This ST and TOE is conformant to EAL2 package claims augmented with ALC_FLR.1.

3.7 Conformance Claim Rationale

There is no Conformance Claim rationale for this ST.

4 Security Problem Definition

4.1 Threats

The following are threats identified for the TOE and the IT System the TOE monitors. The TOE itself has threats and the TOE is also responsible for addressing threats to the environment in which it resides. The assumed level of expertise of the attacker for all the threats is unsophisticated.

4.2 TOE Threats

T.ACCESS A legitimate user of the TOE could gain unauthorized access to resources or information protected by the TOE, or performs operations for which no access rights have been granted, via user error, system error, or other actions.

T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.

T.AUDIT_COMPROMISE A malicious user or process may view audit records and/or IT data, cause the records or information to be lost or modified, or prevent future audit records and IT data from being recorded, thus masking a user's action.

T.EAVESDROPPING A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data.

T.INTERFACE_ABUSE A malicious user may attempt to communicate with the TOE's interfaces in ways that exploit flaws, subvert the TOE, or defeat the operation of its security mechanisms.

T.MASK Users whether they be malicious or non-malicious, could gain unauthorized access to the TOE by bypassing identification and authentication countermeasures.

T.MISUSE Users of an IT product the TOE monitors may perform undesirable actions using the IT product in question, whether by performing malicious actions upon the product, by utilizing functions within the IT product that adversely affect the system it interacts with, or by altering the configuration to be insecure.

T.STEALTH A malicious user or process could perform suspicious activities against the TOE or objects in the Operational Environment monitored by the TOE without a TOE user authorized by the Operational Environment becoming aware of this behavior.

4.3 Organizational Security Policies

There are no Organizational Security Policies that apply to the TOE.

Formatted: Font: Bold

4.4 Assumptions

This section contains assumptions regarding the security environment and the intended usage of the TOE.

4.4.1 Personnel Assumptions

A.ADMIN One or more authorized administrators are assigned to install, configure and manage the TOE and the security of the information it contains.

A.NOEVIL Users of the TOE are not careless, wilfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.

A.PATCHES System Administrators exercise due diligence to update the TOE with the latest patches and patch the Operational Environment (e.g., OS and database) so they are not susceptible to network attacks.

4.4.2 Connectivity Assumptions

A.FILESYS The security features offered by the Operational Environment protect the files used by the TOE.

A.SECURE The network which the TOE is monitoring is expected to be secure to protect the transfer of data from remote data sources and email messages sent to the SMTP Server.

4.4.3 Physical Assumptions

A.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.

5 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.

5.1 IT Security Objectives

The following are the TOE security objectives:

O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.

O.ALERT The TOE will provide measures for determining security alerts when audit data or IT records that represent any of these alerts is recorded.

O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE, the information it collects, and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.

O.AUTH The TOE will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the TOE.

O.EAVESDROPPING The TOE will encrypt TSF data that traverses the network to prevent malicious users from gaining unauthorized access to TOE data.

O.INDEX The TOE will provide measures for collecting security relevant events from configured IT products. These events that will assist the authorized users in detecting misuse of the IT products, the information they collect, and/or their security features that would compromise the integrity of the IT product and violate the security objectives of the IT product.

O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.

O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.

O.SELF_PROTECTION The TOE will preserve a secure state even in the presence of adversarial activity and ensure overall TOE functionality remains operational when a component of the TOE fails.

5.2 Security Objectives for the operational environment of the TOE

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

OE.ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.

OE.AUDIT The Operational Environment will provide local access control, storage of the event audit records which are stored on the machine where the TOE is installed.

OE.FILESYS The security features offered by the Operational Environment will protect the files used by the TOE.

OE.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.

OE.NOEVIL All users of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.

OE.SECURE The network which the TOE is monitoring will be secured to protect data being sent to the TOE as well as SMTP messages sent from the TOE.

OE.SYSTIME The Operational Environment will provide reliable system time.

6 Extended Security Functional and Assurance Requirements

6.1 Extended Security Functional Requirements for the TOE

The following table provides a summary of the Security Functional Requirements that are implemented by the TOE.

Security Function	Security Functional Components
IT Data Indexing (FAU_EXT)	FAU_GEN_EXT.1 IT Data Collection
	FAU_SAR_EXT.1 IT Data Review
	FAU_SAR_EXT.2 Restricted IT data review
	FAU_SAR_EXT.3 Selectable IT Data Review
	FAU_STG_EXT.2 Guarantees of IT Data Availability
	FAU_STG_EXT.3 Action in Case of IT Data Loss
	FAU_STG_EXT.4 Prevention of IT Data Loss

Table 6-1: Extended Security Functional Requirements for the TOE

6.1.1 Class FAU_EXT: IT Data Indexing

6.1.1.1 FAU_GEN_EXT.1 Component Definition

The purpose of creating additional requirements for data collection is to highlight the primary functionality of the TOE. There are no current SFRs that refer to IT data collection or generic data collection. The closest requirements available are the Security Audit class of requirements. These are slightly altered to pertain to IT data rather than audit records.

Hierarchical to: No other components.

FAU_GEN_EXT.1.1 The TSF shall be able to monitor and/or collect the following information from targeted IT resource(s): **[*assignment: list of IT resources***].

FAU_GEN_EXT.1.2 The TSF shall record within each IT data record at least the following information: [***assignment: list of IT data information collected***].

Dependencies: FPT_STM.1 Reliable time stamps

6.1.1.2 FAU_GEN_EXT.1 IT Data Collection

Hierarchical to: No other components.

FAU_GEN_EXT.1.1 The TSF shall be able to monitor and/or collect the following information from targeted IT resource(s): **[*Windows event logs, TCP syslog, UDP syslog, Local disk text files, Generic scripted, Active Directory monitoring (local), Windows registry changes (Regmon), File system changes***].

FAU_GEN_EXT.1.2 The TSF shall record within each IT data record at least the following information: [***Raw log data, Date and time of the event, Host, Line Count, Punctuation, Source, Source type***].

Dependencies: FPT_STM.1 Reliable time stamps

6.1.1.3 FAU_SAR_EXT.1 Component Definition

The purpose of creating additional requirements for data review is to highlight the primary functionality of the TOE. There are no current SFRs that refer to IT data review or generic data review. The closest requirements available are the Security Audit class of requirements. These are slightly altered to pertain to IT data rather than audit records.

Hierarchical to: No other components.

FAU_SAR_EXT.1.1 The TSF shall provide [***assignment: authorized users***] with the capability to read [***assignment: list of IT data information***] from the IT data records.

FAU_SAR_EXT.1.2 The TSF shall provide the IT data records in a manner suitable for the user to interpret the information.

Dependencies: FAU_GEN_EXT.1 IT Data Collection

6.1.1.4 FAU_SAR_EXT.1 IT Data Review

Hierarchical to: No other components.

FAU_SAR_EXT.1.1 The TSF shall provide [***authorized users***] with the capability to read [***all data collected in FAU_GEN_EXT.1***] from the IT data records.

FAU_SAR_EXT.1.2 The TSF shall provide the IT data records in a manner suitable for the user to interpret the information.

Dependencies: FAU_GEN_EXT.1 IT Data Collection

6.1.1.5 FAU_SAR_EXT.2 Component Definition

Hierarchical to: No other components.

FAU_SAR_EXT.2.1 The TSF shall prohibit all users read access to the IT data records, except those users that have been granted explicit read-access.

Dependencies: FAU_SAR_EXT.1 IT Data review

6.1.1.6 FAU_SAR_EXT.2 Restricted IT data review

Hierarchical to: No other components.

FAU_SAR_EXT.2.1 The TSF shall prohibit all users read access to the IT data records, except those users that have been granted explicit read-access.

Dependencies: FAU_SAR_EXT.1 IT Data review

6.1.1.7 FAU_SAR_EXT.3 Component Definition

Hierarchical to: No other components.

FAU_SAR_EXT.3.1 The TSF shall provide the ability to apply [***assignment: methods of selection and/or ordering***] of IT data based on [***assignment: criteria with logical relations***].

Dependencies: FAU_SAR_EXT.1 IT Data review

6.1.1.8 FAU_SAR_EXT.3 Selectable IT Data Review

Hierarchical to: No other components.

FAU_SAR_EXT.3.1 The TSF shall provide the ability to apply [***searching***] of IT data based on [***all data collected in FAU_GEN_EXT.1***].

Dependencies: FAU_SAR_EXT.1 IT Data review

6.1.1.9 FAU_STG_EXT.2 Component Definition

The purpose of creating additional requirements for data storage is to highlight the primary functionality of the TOE. There are no current SFRs that refer to IT data storage or generic data storage. The closest requirements available are the Security Audit class of requirements. These are slightly altered to pertain to IT data rather than audit records.

Hierarchical to: No other components.

FAU_STG_EXT.2.1 The TSF shall protect the stored IT data records in the IT data trail from unauthorised deletion.

FAU_STG_EXT.2.2 The TSF shall be able to [selection, choose one of: prevent, detect] unauthorised modifications to the stored IT data records in the IT data trail.

FAU_STG_EXT.2.3 The TSF shall ensure that [***assignment: metric for saving IT data records***] stored IT data records will be maintained when the following conditions occur: [selection: IT data storage exhaustion, failure, attack].

Dependencies: FAU_GEN_EXT.1 IT Data Collection

6.1.1.10 FAU_STG_EXT.2 Guarantees of IT Data Availability

Hierarchical to: No other components.

FAU_STG_EXT.2.1 The TSF shall protect the stored IT data records in the IT data trail from unauthorised deletion.

FAU_STG_EXT.2.2 The TSF shall be able to [prevent] unauthorised modifications to the stored IT data records in the IT data trail.

Application Note: *These requirements are satisfied intrinsically because the TOE does not provide any mechanism for performing any modification or deletion to stored IT data. However, the can_delete role and its associated capability has the ability to delete the index pointers to the data. This does not delete the actual data from the machine, and a new index can be created.*

FAU_STG_EXT.2.3 The TSF shall ensure that [***the most recent***] stored IT data records will be maintained when the following conditions occur: [IT data storage exhaustion].

Dependencies: FAU_GEN_EXT.1 IT Data Collection

6.1.1.11 FAU_STG_EXT.3 Component Definition

Hierarchical to: No other components.

FAU_STG_EXT.3.1 The TSF shall [***assignment: actions to be taken in case of possible IT data storage failure***] if the IT data trail exceeds [***assignment: pre-defined limit***].

Dependencies: FAU_GEN_EXT.1 IT Data Collection

6.1.1.12 FAU_STG_EXT.3 Action in Case of IT Data Loss

Hierarchical to: No other components.

FAU_STG_EXT.3.1 The TSF shall [***stop indexing***] if the IT data trail exceeds [***IT data storage exhaustion***].

Dependencies: FAU_GEN_EXT.1 IT Data Collection

6.1.1.13 FAU_STG_EXT.4 Component Definition

Hierarchical to: FAU_STG_EXT.3 Action in Case of IT Data Loss

FAU_STG_EXT.4.1 The TSF shall [selection, choose one of: "ignore IT data events", "prevent IT data events, except those taken by the authorised user with special rights", "overwrite the oldest stored IT data records"] and [***assignment: other actions to be taken in case of IT data storage failure***] if the IT data trail is full.

Dependencies: FAU_STG_EXT.2 Guarantees of IT Data Availability

6.1.1.14 FAU_STG_EXT.4 Prevention of IT Data Loss

Hierarchical to: FAU_STG_EXT.3 Action in Case of IT Data Loss

FAU_STG_EXT.4.1 The TSF shall [overwrite the oldest stored IT data records] and [***send an alert***] if the IT data trail is full.

Dependencies: FAU_STG_EXT.2 Guarantees of IT Data Availability

6.2 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

7 Security Functional Requirements

7.1 Security Functional Requirements for the TOE

The following table provides a summary of the Security Functional Requirements implemented by the TOE.

Security Function	Security Functional Components
Security Audit (FAU)	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR.2 Restricted audit review
	FAU_SAR.3 Selectable audit review
	FAU_STG.2 Guarantees of audit data availability
	FAU_STG.3 Action in case of audit data loss
Cryptographic Support (FCS)	FCS_CKM.1 Cryptographic key generation
	FCS_CKM.4 Cryptographic key destruction
	FCS_COP.1 Cryptographic operation
User Data Protection (FDP)	FDP_ACC.1 Access control policy
	FDP_ACF.1 Access control functions
Identification and Authentication (FIA)	FIA_ATD.1 User attribute definition
	FIA_UAU.2 User authentication before any action
	FIA_UAU.5 Multiple authentication mechanisms
	FIA_UAU.6 Re-authenticating
	FIA_UID.2 User identification before any action
	FIA_USB.1 User-subject binding
Security Management (FMT)	FMT_MOF.1 Management of security functions behavior
	FMT_MSA.1 Management of security attributes
	FMT_MSA.3 Static attribute initialization
	FMT_MTD.1 Management of TSF data
	FMT_REV.1 Revocation
	FMT_SMF.1 Specification of Management Functions
	FMT_SMR.1 Security roles
Protection of the TSF (FPT)	FPT_FLS.1 Failure with preservation of secure state
	FPT_ITC.1 Confidentiality of exported TSF data
	FPT_ITI.1 Integrity of exported TSF data
	FPT_ITT.1 Basic internal TSF data transfer protection
Resource Utilization (FRU)	FRU_FLT.1 Fault tolerance
Trusted Path/Channels (FTP)	FTP_TRP.1 Trusted path

Table 7-1: Security Functional Requirements for the TOE

7.1.1 Class FAU: Security Audit

7.1.1.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

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

Start-up and shutdown of the audit functions;

All auditable events for the [not specified] level of audit; and

[***assignment: other specifically defined auditable events***].

Component	Event
FAU_GEN_EXT.1	None
FAU_SAR_EXT.1	Searching IT data
FAU_STG_EXT.2	Actions taken due to exceeding the IT data threshold
FAU_STG_EXT.3	Actions taken due to exceeding the IT data threshold
FAU_STG_EXT.4	Actions taken due to exceeding the IT data threshold
FAU_GEN.1	None
FAU_GEN.2	None
FAU_SAR.1	Searching audit data
FAU_SAR.2	Searching audit data
FAU_SAR.3	Searching audit data
FAU_STG.2	Actions taken due to exceeding the audit data threshold
FAU_STG.3	Actions taken due to exceeding the audit data threshold
FCS_CKM.1	None
FCS_CKM.4	None
FCS_COP.1	None
FDP_ACC.1	None
FDP_ACF.1	None
FIA_ATD.1	None
FIA_UAU.2	Successful and unsuccessful use of authentication mechanisms
FIA_UAU.5	Successful and unsuccessful use of authentication mechanisms
FIA_UAU.6	Successful and unsuccessful use of authentication mechanisms
FIA_UID.2	Successful and unsuccessful use of authentication mechanisms
FIA_USB.1	Success and failure of binding of user security attributes to a subject
FMT_MOF.1	All modifications in the behavior of the functions in the TSF (See Table 7-3)

FMT_MSA.1	All manipulation of the security attributes (See Table 7-4)
FMT_MSA.3	None
FMT_MTD.1	All modifications of the values of TSF data by the administrator (See Table 7-3)
FMT_REV.1	All attempts to revoke security attributes (See Table 7-4)
FMT_SMF.1	All use of the management functions (See Table 7-3)
FMT_SMR.1	Modifications of users assigned to a role
FPT_FLS.1	None
FPT_ITC.1	None
FPT_ITI.1	None
FPT_ITT.1	None
FRU_FLT.1	None
FTP_TRP.1	All attempted uses of the trusted path functions

Table 7-2: Auditable Events

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [***audit ID, hash signature***].

Application Note: *Audit logs are generated as a local file (audit.log) and are also added to the index.*

Dependencies: FPT_STM.1 Reliable time stamps

7.1.1.2 FAU_GEN.2 User identity association

Hierarchical to: No other components.

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

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

7.1.1.3 FAU_SAR.1 Audit review

Hierarchical to: No other components.

FAU_SAR.1.1 The TSF shall provide [***authorized users***] with the capability to read [***all information collected by FAU_GEN.1***] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

Application Note: *Audit logs are read using the local system's read mechanism. The term authorized users means any user with a role that contains the search capability and has access to the _audit index.*

Dependencies: FAU_GEN.1 Audit data generation

7.1.1.4 FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

Dependencies: FAU_SAR.1 Audit review

7.1.1.5 FAU_SAR.3 Selectable Audit Review

Hierarchical to: No other components.

FAU_SAR.3.1 The TSF shall provide the ability to apply [***sort and filter***] of audit data based on [***one or more of the following: Date and time of the event, Type of event, Subject identity (if applicable), The outcome (success/failure) of the event***].

Dependencies: FAU_SAR.1 Audit review

7.1.1.6 FAU_STG.2 Guarantees of audit data availability

Hierarchical to: FAU_STG.1 Protected audit trail storage

FAU_STG.2.1 The TSF shall protect the stored audit records from unauthorized deletion.

FAU_STG.2.2 The TSF shall be able to [prevent] unauthorized modifications to the stored audit records in the audit trail.

FAU_STG.2.3 The TSF shall ensure that [***the most recent***] stored audit records will be maintained when the following conditions occur: [audit storage exhaustion].

Application Note: *The TOE disables user actions when audit storage is exhausted. This is because audit storage has no set cap. Therefore, the cap of audit storage is the size of the hard disk. All user actions must be logged to be performed.*

Dependencies: FAU_GEN.1 Audit data generation

7.1.1.7 FAU_STG.3 Action in case of audit data loss

Hierarchical to: No other components.

FAU_STG.3.1 The TSF shall [***stop indexing, but allow auditable events***] if the audit trail exceeds [***configurable free space minimum***].

Dependencies: FAU_GEN.1 Audit data generation

7.1.1.8 FAU_STG.4 Prevention of audit data loss

Hierarchical to: FAU_STG.3 Action in case of possible audit data loss

FAU_STG.4.1 The TSF shall [overwrite the oldest stored audit records] and [***take no other action***] if the audit trail is full.

Application Note: *Audit records in this case refer to only the indexed audit records. The audit records in audit.log are not deleted using this functionality. This is the same functionality as the rollover function for IT data.*

Dependencies: FAU_STG.1 Protected audit trail storage

7.1.2 Class FCS: Cryptographic Support

7.1.2.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to: No other components.

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [***RSA***] and specified cryptographic key sizes [***1024 bit***] that meet the following: [***RFC 3565***].

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or

FCS_COP.1 Cryptographic operation]

FCS_CKM.4 Cryptographic key destruction

7.1.2.2 FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [***overwrite***] that meets the following: [***no standard***].

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

7.1.2.3 FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform [***encryption and decryption***] in accordance with a specified cryptographic algorithm [***AES***] and cryptographic key sizes [***1024 bit***] that meet the following: [***RFC 3565***].

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

7.1.3 Class FDP: User Data Protection

7.1.3.1 FDP_ACC.1 Access control policy

Hierarchical to: No other components.

FDP_ACC.1.1 The TSF shall enforce the [***RBAC Policy***] on [***users attempting to query TSF data or configure its operation***].

Dependencies: FDP_ACF.1 Security attribute based access control

7.1.3.2 FDP_ACF.1 Access control functions

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the [***RBAC Policy***] to objects based on the following: [

- ***the capabilities assigned to a role***
- ***the indexes assigned to a role***
- ***the roles assigned to an object's ACL***

- ***the roles assigned to a user***].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [***see Table 7-3 below***].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [***if a user is assigned a role with the "Admin All Objects" capability, ACLs are no longer considered when granting access***].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [***none***].

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

7.1.4 Class FIA: Identification and Authentication

7.1.4.1 FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [***username, password, roles***].

Dependencies: No dependencies.

7.1.4.2 FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: FIA_UID.1 Timing of identification

7.1.4.3 FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components.

FIA_UAU.5.1 The TSF shall provide [***Splunk authentication, LDAP authentication***] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [***configuration assigned by an authorized user***].

Dependencies: No dependencies.

7.1.4.4 FIA_UAU.6 Re-authentication

Hierarchical to:	No other components.
FIA_UAU.6.1	The TSF shall re-authenticate the user under the conditions [***session inactivity***].
Dependencies:	No dependencies.

7.1.4.5 FIA_UID.2 User identification before any action

Hierarchical to:	FIA_UID.1 Timing of identification
FIA_UID.2.1	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	No dependencies.

7.1.4.6 FIA_USB.1 User-subject binding

Hierarchical to:	No other components.
FIA_USB.1.1	The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [***role***].
FIA_USB.1.2	The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [***association of a user's session and roles available to the user***].
FIA_USB.1.3	The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [***Revocation of the user-subject binding and termination of the user's session under one of the following conditions: The user is deleted, All roles have been removed for the user***].
Dependencies:	FIA_ATD.1 User attribute definition

7.1.5 Class FMT: Security Management

7.1.5.1 FMT_MOF.1 Management of security functions behavior

Hierarchical to:	No other components.
FMT_MOF.1.1	The TSF shall restrict the ability to [***see Action column in Table 7-3 below***] the functions [***see Object column in Table 7-3 below***] to [***users which possess the items in the Capability Applicable and ACL Applicable columns in Table 7-3 below***].

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

Object	Action	Capability Applicable?	ACL Applicable?	Index Applicable?
Index Data	search allowed indexes	search	no	yes
	search allowed indexes in real time	rtsearch	no	yes
	typeahead into allowed indexes	get_typeahead	no	yes
	hide specific events with 'delete'	delete_by_keyword	no	yes
	get metadata about allowed indexes	get_metadata	no	yes
Search Jobs	list/cancel/pause/etc.	no	yes	yes
LDAP-auth (including editing groups)	create, modify, query	change_authentication	no	no
auth-services (get current auth system type)	list	change_authentication	no	no
auth-tokens	create/list	no (protected by public/private keys)	no	no
capabilities	list	no	no	no
commands (external python commands, like crawl)	list	no	yes	no
conf-inputs, conf-wmi	list	list_inputs	no	no
current-context	get current user context	no	no	no
deploymentclient	edit/list	edit_deployment_client	no	no
deploymentserver, deploymentserverclass, tenants	create/edit/list	edit_deployment_server	no	no
directory (list admin handlers)	list	no	no	no
distsearch-peer	create/list	edit_dist_peer	no	no

Object	Action	Capability Applicable?	ACL Applicable?	Index Applicable?
distsearch-setup	create/list	edit_search_server	no	no
eventtypes	create/edit/list	no	yes	no
fieldaliases	create/edit/list	no	yes	no
httpauth-tokens (access user sessions)	end session	edit_httpauths	no	no
	list	list_httpauths	no	no
indexes	create/edit	indexes_edit	no	no
	list	no	no	no
inputstatus	list	no	no	no
license	create/edit/list	license_tab	no	no
logger (logging levels)	edit	edit_server	no	no
	list	no	no	no
lookup-table-files	create/edit/list	no	yes	no
macros	create/edit/list	no	yes	no
manager, nav, views, viewstates (UI page access)	create/edit/list	no	yes	no
oneshotinput	create/edit	edit_monitor	no	no
	list	list_inputs	no	no
props-extract (field extractions)	create/edit/list	no	yes	no
props-lookup (lookups in props.conf)	create/edit/list	no	yes	no
roles	create/edit	edit_roles	no	no
	list	no	no	no
savedsearch	create/edit/list	no	yes	yes
scheduledviews	edit/list	no	yes	yes
script (scripted inputs)	create/edit	edit_scripted	no	no
	list	list_inputs	no	no
server-control	restart splunk	restart_splunkd	no	no
server-info	list	no	no	no

Object	Action	Capability Applicable?	ACL Applicable?	Index Applicable?
server-settings	list	edit_server	no	no
	change default host for inputs	edit_input_defaults	no	no
	change server settings	edit_server	no	no
	change splunkweb settings	edit_web_settings	no	no
sourcetype	list	no	no	no
sourcetype-rename	create/edit/list	no	yes	no
syslog (output)	create/edit	edit_forwarders	no	no
	list	list_forwarders	no	no
tags	create/edit/list	no	yes	no
TCP input - raw	create/edit	edit_tcp	no	no
	list	list_inputs	no	no
TCP input - cooked	create/edit	edit_splunktcp	no	no
	list	list_inputs	no	no
tcpout-default, tcpout-group, tcpout-server	create/edit	edit_forwarders	no	no
	list	list_forwarders	no	no
transforms-extract, transforms-lookup	create/edit/list	no	yes	no
udp (input)	create/edit	edit_udp	no	no
	list	list_inputs	no	no
user-prefs	create/edit/list	no	no	no
users	create/edit	edit_user	no	no
	change role	edit_user	no	no
	change own password	change_own_password	no	no
	list	edit_user required to see all users	no	no
workflow-actions	create/edit/list	no	yes	no
win-eventlogs	create/edit	edit_win_eventlogs	no	no

Object	Action	Capability Applicable?	ACL Applicable?	Index Applicable?
	list	list_inputs	no	no
win-wmi-wql	create/edit	edit_win_wmiwql	no	no
	list	list_inputs	no	no
win-regmon	create/edit	edit_win_regmon	no	no
	list	list_inputs	no	no
win-admon	create/edit	edit_win_admon	no	no
	list	list_inputs	no	no

Table 7-3: Management Functions of the TOE

7.1.5.2 FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the [***RBAC Policy***] to restrict the ability to [***See Operations column in table 7-4 below***] the security attributes [***See Object column in table 7-4 below***] to [***users with the Attribute listed in table 7-4 below***].

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

Object	Operation	Attribute
Username	Create, modify, delete, query	Edit User Capability
Role	Create, modify, delete, query	Edit Role Capability
Capability	***Assign***	Edit Role Capability
Index	***Assign***	Edit Role Capability
ACL	Create, modify, delete, query	Object Ownership

Table 7-4: Assignment of Security Attributes

7.1.5.3 FMT_MSA.3 Static attribute initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the [***RBAC Policy***] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [***users with attribute listed in the Attributes column in Table 7-4 above***] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

7.1.5.4 FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to [***see the Actions column in Table 7-3 above***] the [***see the Objects column in Table 7-3 above***] to [***users which possess the items in the Capability Applicable and ACL Applicable columns in Table 7-3 above***].

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

7.1.5.5 FMT_REV.1 Revocation

Hierarchical to: No other components.

FMT_REV.1.1 The TSF shall restrict the ability to revoke [***username and role***] associated with the [users] under the control of the TSF to [***authorized users***].

FMT_REV.1.2 The TSF shall enforce the rules [

a) Revocation of the user-subject binding and termination of the user's session under one of the following conditions: The user is deleted, All roles have been removed for the user

b) Updating of the user-subject binding when a user's role(s) change].

Application Note: *When roles are changed within the TOE, the module in charge of authorization will update the user to role binding.*

Dependencies: FMT_SMR.1 Security roles

7.1.5.6 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [***see table 7-3 above***].

Dependencies: No dependencies.

7.1.5.7 FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles [***admin, can_delete, power, user, and any user defined roles***].

Application Note: *The can_delete role only contains the capability to delete data by keyword. This data is not deleted, but hidden from searches. No other role contains this ability. Typical usage of the can_delete role is to add it to a user in addition to their regular role to grant the delete by keyword functionality.*

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Dependencies: FIA_UID.1 Timing of identification

7.1.6 Class FPT: Protection of the TOE Security Functions

7.1.6.1 FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [***the failure of an indexer in an auto-balanced set of indexers***].

Dependencies: No dependencies.

7.1.6.2 FPT_ITC.1 Confidentiality of Exported TSF Data

Hierarchical to: No other components.

FPT_ITC.1.1 The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product from unauthorized disclosure during transmission.

Application Note: *The transmission of TSF data from the TOE to other products is protected by OpenSSL.*

Dependencies: No dependencies.

7.1.6.3 FPT_ITI.1 Integrity of Exported TSF Data

Hierarchical to: No other components.

FPT_ITI.1.1 The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and another trusted IT product within the following metric: [***OpenSSL 0.9.8n***].

FPT_ITI.1.2 The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and another trusted IT product and perform [***a dropping of the packet***] if modifications are detected.

Application Note: *The transmission of TSF data from the TOE to other products is protected by OpenSSL.*

Dependencies: No dependencies.

7.1.6.4 FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

FPT_ITT.1.1 The TSF shall protect TSF data from [disclosure, modification] when it is transmitted between separate parts of the TOE.

Application Note: *The transmission of TSF data between TOE components is protected by OpenSSL.*

Dependencies: No dependencies.

7.1.7 Class FRU: Resource Utilization

7.1.7.1 FRU_FLT.1 Fault Tolerance

Hierarchical to: No other components.

FRU_FLT.1.1 The TSF shall ensure the operation of [***IT data indexing***] when the following failures occur: [***the failure of an indexer in an auto-balanced set of indexers***].

Dependencies: FPT_FLS.1 Failure with preservation of secure state

7.1.8 Class FTP: Trusted Paths/Channels

7.1.8.1 FTP_TRP.1 Trusted Paths

Hierarchical to: No other components.

FTP_TRP.1.1 The TSF shall provide a communication path between itself and [remote] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [modification, disclosure].

FTP_TRP.1.2	The TSF shall permit [remote users] to initiate communication via the trusted path.
FTP_TRP.1.3	The TSF shall require the use of the trusted path for [initial user authentication, management actions, data transfer].
Dependencies:	No dependencies.

7.2 Operations Defined

The requirements in this document are divided into assurance requirements and two sets of functional requirements. The first set of functional requirements, which were drawn from the Common Criteria, is designed to address the core System requirements for self-protection. The second set of requirements, which were modified from existing Security Audit (FAU) requirements, are designed to address the requirements for the TOE's primary function, which is collection and indexing of IT data and the ability to search said data.

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

- Assignment: allows the specification of an identified parameter. Indicated with ***bold text and italics*** if further operations are necessary by the Security Target author;
- Refinement: allows the addition of details. Indicated with ***bold text and italics*** if further operations are necessary by the Security Target author;
- Selection: allows the specification of one or more elements from a list. Indicated with underlined text; and
- Iteration: allows a component to be used more than once with varying operations.

8 Security Assurance Requirements

This section identifies the Security Assurance Requirement components met by the TOE. These assurance components meet the requirements for EAL2 augmented with ALC_FLR.1.

8.1 Security Architecture

8.1.1 Security Architecture Description (ADV_ARC.1)

ADV_ARC.1.1D: The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D: The developer shall design and implement the TSF so that it is able to protect itself from tampering by un-trusted active entities.

ADV_ARC.1.3D: The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1C: The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C: The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C: The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4C: The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C: The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

ADV_ARC.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.1.2 Security-enforcing functional specification (ADV_FSP.2)

ADV_FSP.2.1D: The developer shall provide a functional specification.

ADV_FSP.2.2D: The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.2.1C: The functional specification shall completely represent the TSF.

ADV_FSP.2.2C: The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3C: The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4C: For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.2.5C: For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6C: The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E: The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

8.1.3 Basic Design (ADV_TDS.1)

ADV_TDS.1.1D: The developer shall provide the design of the TOE.

ADV_TDS.1.2D: The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.1.1C: The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.1.2C: The design shall identify all subsystems of the TSF.

ADV_TDS.1.3C: The design shall describe the behavior of each SFR-supporting or SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.

ADV_TDS.1.4C: The design shall summarise the SFR-enforcing behavior of the SFR-enforcing subsystems.

ADV_TDS.1.5C: The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.

ADV_TDS.1.6C: The mapping shall demonstrate that all TSFIs trace to the behavior described in the TOE design that they invoke.

ADV_TDS.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.1.2E: The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

8.2 Guidance Documents

8.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1D	The developer shall provide operational user guidance.
AGD_OPE.1.1C	The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
AGD_OPE.1.2C	The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.
AGD_OPE.1.3C	The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.
AGD_OPE.1.4C	The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_OPE.1.5C	The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AGD_OPE.1.6C	The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.
AGD_OPE.1.7C	The operational user guidance shall be clear and reasonable.
AGD_OPE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1D	The developer shall provide the TOE including its preparative procedures.
AGD_PRE.1.1C	The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
AGD_PRE.1.2C	The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of

the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator *shall apply* the preparative procedures to confirm that the TOE can be prepared securely for operation.

8.3 Lifecycle Support

8.3.1 Use of a CM system (ALC_CMC.2)

ALC_CMC.2.1D: The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.2.2D: The developer shall provide the CM documentation.

ALC_CMC.2.3D: The developer shall use a CM system. ALC_CMC.2.1C: The TOE shall be labeled with its unique reference.

ALC_CMC.2.2C: The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.2.3C: The CM system shall uniquely identify all configuration items.

ALC_CMC.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.2 Parts of the TOE CM coverage (ALC_CMS.2)

ALC_CMS.2.1D: The developer shall provide a configuration list for the TOE.

ALC_CMS.2.1C: The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.

ALC_CMS.2.2C: The configuration list shall uniquely identify the configuration items.

ALC_CMS.2.3C: For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

ALC_CMS.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.3 Delivery Procedures (ALC_DEL.1)

ALC_DEL.1.1D The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

ALC_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.4 Flaw reporting procedures (ALC_FLR.1)

ALC_FLR.1.1D The developer shall document and provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.1.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.1.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4 Security Target Evaluation

8.4.1 Conformance Claims (ASE_CCL.1)

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C	The CC conformance claim shall be consistent with the extended components definition.
ASE_CCL.1.5C	The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.
ASE_CCL.1.6C	The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
ASE_CCL.1.7C	The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.
ASE_CCL.1.8C	The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

8.4.2 Extended Components Definition (ASE_ECD.1)

ASE_ECD.1.1D	The developer shall provide a statement of security requirements.
ASE_ECD.1.2D	The developer shall provide an extended components definition.
ASE_ECD.1.1C	The statement of security requirements shall identify all extended security requirements.
ASE_ECD.1.2C	The extended components definition shall define an extended component for each extended security requirement.
ASE_ECD.1.3C	The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.
ASE_ECD.1.4C	The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.
ASE_ECD.1.5C	The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.
ASE_ECD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_ECD.1.2E	The evaluator shall confirm that no extended component can be clearly expressed using existing components.

8.4.3 ST Introduction (ASE_INT.1)

ASE_INT.1.1D	The developer shall provide an ST introduction.
ASE_INT.1.1C	The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
ASE_INT.1.2C	The ST reference shall uniquely identify the ST.
ASE_INT.1.3C	The TOE reference shall identify the TOE.
ASE_INT.1.4C	The TOE overview shall summarize the usage and major security features of the TOE.
ASE_INT.1.5C	The TOE overview shall identify the TOE type.
ASE_INT.1.6C	The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.
ASE_INT.1.7C	The TOE description shall describe the physical scope of the TOE.
ASE_INT.1.8C	The TOE description shall describe the logical scope of the TOE.
ASE_INT.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_INT.1.2E	The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

8.4.4 Security objectives (ASE_OBJ.2)

ASE_OBJ.2.1D	The developer shall provide a statement of security objectives.
ASE_OBJ.2.2D	The developer shall provide security objectives rationale.
ASE_OBJ.2.1C	The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.
ASE_OBJ.2.2C	The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.
ASE_OBJ.2.3C	The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.
ASE_OBJ.2.4C	The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C	The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.
ASE_OBJ.2.6C	The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.
ASE_OBJ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.5 Derived security requirements (ASE_REQ.2)

ASE_REQ.2.1D	The developer shall provide a statement of security requirements.
ASE_REQ.2.2D	The developer shall provide a security requirement's rationale.
ASE_REQ.2.1C	The statement of security requirements shall describe the SFRs and the SARs.
ASE_REQ.2.2C	All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.
ASE_REQ.2.3C	The statement of security requirements shall identify all operations on the security requirements.
ASE_REQ.2.4C	All operations shall be performed correctly.
ASE_REQ.2.5C	Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.
ASE_REQ.2.6C	The security requirements rationale shall trace each SFR back to the security objectives for the TOE.
ASE_REQ.2.7C	The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.
ASE_REQ.2.8C	The security requirements rationale shall explain why the SARs were chosen.
ASE_REQ.2.9C	The statement of security requirements shall be internally consistent.
ASE_REQ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.6 Security Problem Definition (ASE_SPD.1)

ASE_SPD.1.1D	The developer shall provide a security problem definition.
ASE_SPD.1.1C	The security problem definition shall describe the threats.

ASE_SPD.1.2C	All threats shall be described in terms of a threat agent, an asset, and an adverse action.
ASE_SPD.1.3C	The security problem definition shall describe the OSPs.
ASE_SPD.1.4C	The security problem definition shall describe the assumptions about the operational environment of the TOE.
ASE_SPD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.7 TOE Summary Specification (ASE_TSS.1)

ASE_TSS.1.1D	The developer shall provide a TOE summary specification.
ASE_TSS.1.1C	The TOE summary specification shall describe how the TOE meets each SFR.
ASE_TSS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_TSS.1.2E	The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

8.5 Tests

8.5.1 Evidence of Coverage (ATE_COV.1)

ATE_COV.1.1D:	The developer shall provide evidence of the test coverage.
ATE_COV.1.1C:	The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.
ATE_COV.1.1E:	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.5.2 Functional Testing (ATE_FUN.1)

ATE_FUN.1.1D	The developer shall test the TSF and document the results.
ATE_FUN.1.2D	The developer shall provide test documentation
ATE_FUN.1.1C	The test documentation shall consist of test plans, expected test results and actual test results.
ATE_FUN.1.2C	The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.3C	The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C	The actual test results shall be consistent with the expected test results.
ATE_FUN.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.5.3 Independent Testing - Sample (ATE_IND.2)

ATE_IND.2.1D	The developer shall provide the TOE for testing.
ATE_IND.2.1C	The TOE shall be suitable for testing.
ATE_IND.2.2C	The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2E	The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.
ATE_IND.2.3E	The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

8.6 Vulnerability Assessment

8.6.1 Vulnerability Analysis (AVA_VAN.2)

AVA_VAN.2.1D	The developer shall provide the TOE for testing.
AVA_VAN.2.1C	The TOE shall be suitable for testing.
AVA_VAN.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VAN.2.2E	The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.
AVA_VAN.2.3E	The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.
AVA_VAN.2.4E	The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

9 TOE Summary Specification

9.1 TOE Security Functions

The following sections identify the security functions of the TOE. They include IT Data Indexing, Security Audit, Cryptographic Support, User Data Protection, Identification and Authentication, Security Management, Protection of the TSF, High Availability, and Security Architecture.

9.1.1 IT Data Indexing

The primary functionality of the TOE is to collect and index IT data from the following log sources:

- Windows event logs: The native Windows API is used for collecting local event logs.
- TCP syslog
- UDP syslog
- Local disk text files
- Generic scripted logs
- Active Directory monitoring (local)
- Windows registry changes: These logs aren't collected by the TOE; there is a Splunk process that actively monitors the Windows OS registry and creates logs of all changes.
- File system changes: These logs aren't collected by the TOE; there is a Splunk process that actively monitors the OS filesystem and creates logs of all changes.

The TOE utilizes methods to parse raw logs from time-series data (data with a timestamp). The utilized indexing methods parse and translate the raw log data into a set of fields that are recorded by the TOE on disk, and associate this information with an index. All authorized users will be able to read the indexed IT data by performing searches within the TOE. The primary mechanism for reviewing IT data within Splunk is the search functionality. The fields are populated by parsing raw IT data logs. Every parsed log includes the following types of data:

- Internal Fields: These fields hold the raw data from the original log.
- Index-time Fields: These fields hold data that is extracted from the raw logs upon indexing.
- Search-time Fields: These fields do not actually hold data within the system. They are formulated upon search. For instance, if you count the number of login

requests and display that metric, that information is evident without having the statistic in the original log.

- Datetime Fields: These fields allow users to search by independent elements in the event's timestamp. They are formulated at index-time from the parsed timestamp.

A complete list of all default fields within a Splunk deployment are as follows:

Internal Fields	_raw	The original raw data of an event. Contains the entire event log prior to Splunk processing and indexing.
	_time	The event's timestamp expressed in Unix time. This field is formulated from the collective datetime fields (date_hour, date_minute, etc.)
Index-time Fields	host	The originating hostname or IP address of the network device that generated the event.
	linecount	Number of lines an event contains.
	punct	A punctuation pattern that is extracted from an event. These patterns are unique to types of events. The punct field is populated by extracting some limited number of punctuation from the first line of events. The purpose of the punct field is to allow advanced users to be able to classify certain types of event logs by the unique strings of initial punctuation. An example of usage of the punct field in a search is as follows: punct=":*:" This searches for all punctuation patterns that begin and end with colons.
	source	The filename or pathname from which the event was indexed.
	sourcetype	A source classification. Authorized users can define source types, or Splunk can generate them automatically upon indexing.
	timestamp	An event's timestamp value, extracted upon indexing. The format of this field is configurable based on how an authorized user has timestamp extraction configured.
Search-time fields	eventtype	Event types that a user has defined for an event. Users classify events into an event type by finding similar patterns in data, and then saving an event type based on similarities between events. Creating event type involves creating a search. For example, an example of a search to create event type "failed_login" is as follows: "failed login" OR "FAILED LOGIN" OR "Authentication failure" OR "Failed to authenticate user" This will add "failed_login" under the eventtype field for any newly indexed log that contains any of those strings of characters. Another common usage is to utilize the punct command to single out specific event types based upon their unique punctuation strings. More than one value can be set for each indexed log's eventtype field.

	index	The name of the index within which a given event is indexed. By default, all events are indexed in the "main" index.
	splunk-server	The server names in a distributed Splunk environment.
Datetime Fields	date_hour	The value of the hour in which an event occurred. Has a value range of 0-23. This value is extracted from the raw log data.
	date_mday	The value of the day of the month in which an event occurred. Has a value range of 1-31. This value is extracted from the raw log data.
	date_minute	The value of the minute in which an event occurred. Has a value range of 0-59. This value is extracted from the raw log data.
	date_second	The value of the second in which an event occurred. Has a value range of 0-59. This value is extracted from the raw log data.
	date_wday	The value of the day of the week on which an event occurred. Values can be Sunday, Monday, etc. This value is determined by extracting raw log data.and determining what weekday it translates to.
	date_year	The value of the year in which an event occurred. This value is extracted from the raw log data.
	date_zone	The value of time for the local timezone of an event expressed as hours in Unix time. This value is extracted from the raw log data. Utilized to offset an event's timezone by specifying an offset in minutes.

Table 9-1: Fields of Indexed IT Data Logs

A user can also utilize the search API to extract fields in different ways. The following commands can be utilized to extract various fields:

- rex performs field extractions using Perl-compatible regular expressions (PCRE) named groups.
- extract (or kv, for "key/value") explicitly extracts field/values using default patterns.
- multikv extracts field/values on multi-line, tabular-formatted events.
- xmlkv extracts field/values on xml-formatted event data.
- kvform extracts field/values based on predefined form templates.

In addition, the Interactive Field Extractor (IFX) in Splunk Web can be utilized for extracting fields. Therefore, the manner in which Splunk behaves is highly configurable based on a user's level of involvement with configuring Splunk and their ability to create custom regular expressions or patterns.

The search functionality allows users to enter search values for either any text available within the parsed index logs or by specifying values for each and any of the applicable fields within an indexed log. Additionally, for advanced searches, standard search engine logic can be used to further narrow down a search. A user can add "OR," "AND," and

"NOT" as logical operators, and set orders of precedence by surrounding arguments with parentheses. In addition, wildcards can be used by users to further improve the search. For example, searching for "*local*" will return any indexed log that has the word or partial phrase "local" in one of the fields or raw text. Users that can utilize these advanced search features can have more control over the accuracy and precision of information returned by the search functionality.

In addition to the standard search functionality, the TOE also has the ability to save searches. These saved searches are stored in one of two fashions:

- Search conditions: to perform the search again at a later time on a potentially updated index
- Search results: to keep a record of that specific search

Saved searches can also be set to be scheduled, allowing the TOE to automatically perform the same search regularly and save and/or return the results to the search-head.

Alert conditions can be configured for scheduled searches upon definable criteria based upon the extracted fields used in the Splunk installation. In the event of an alert condition being reached or exceeded, the TOE will formulate an email about the occurrence and send it to the configured email recipients. This is done by using an external SMTP server and connecting to it using a third-party SMTP client within the TOE.

All IT data logs indexed are protected from deletion or modification. There are no methods within the TOE that allow users to delete or modify collected data. There is a specific role and related capability within the TOE that gives the ability to flag index pointers as no longer valid, obfuscating the data they point to. However, the raw data is still intact, and there is no mechanism to delete this raw data within the TOE. Additionally, this raw data can also be re-indexed into the system to replace discarded index references. Therefore, if no mechanism exists to permanently delete the collected data, this requirement is satisfied intrinsically.

The most recent stored IT records will be maintained if storage runs out. There is a configurable quota within the TOE that determines the minimum amount of free space left on the hard drive on which Splunk is installed. The default value and value utilized in the evaluated configuration is 2000 MB. In the evaluated configuration, the TOE is configured to stop indexing and begin to overwrite the oldest IT data records when storage reaches the configured limit, which is synonymous with the storage being full. The TOE will also send an alarm in the form of a Splunk Web banner. Once the TOE frees up sufficient space, the indexer will begin indexing once more. In addition to this functionality, the TOE will also be configured to delete IT data logs after they have exceeded a certain age.

9.1.2 Security Audit

The TOE supports security audit generation. The TOE generates audits based on the following actions, as well as the actions listed in Table 7-1:

- Adding, removing, or changing Splunk's configuration files. The TOE monitors for changes using the file system change monitor.
- System start and stop.
- Users logging in and out.
- Execution of any capability in the system.

Audit events that are generated are hashed by utilizing the SHA256 algorithm to generate a hash signature, which is appended to the audit log itself. Within each collected audit data log, the following data exists:

- Timestamp: This contains the date and time of the event
- Action: This contains the action that was requested by the user or system.
- User information: This contains the username of the user who generated the event. If the event describes an action where no user information is captured, this is set to the user whom is currently logged on
- Additional information: This contains the filename of the data with which the user interacted, as well as a determination of whether the action was successful, or if the action was denied.
- ID: This contains a sequential number assigned to the event. The purpose of the ID is for detecting gaps in data.
- Hash signature: This contains a PKI encrypted SHA256 hash signature, which includes the timestamp and ID.

The following is a sample of an audit log entry:

11-01-2007 09:23:59.581 INFO AuditLogger - Audit:[timestamp=Thu Nov 1 09:23:59 2007, id=1, user=admin

All audit events are stored locally in two locations: the audit index (_audit) and the log file (audit.log). When audit data is entered into the audit index, it follows the same rules as the IT data within the index with respect to which fields are extracted. Therefore, the standard Splunk search functionality can be utilized to view audit records.

Formulating a search for audit records utilizes the same search terminology and logical operators as normal IT data searches. If a user wants to return all audit data, that user can perform a search for "index=_audit."

All audits are stored in the _audit index, and therefore any user with a role that is able to view the data in _audit can read all of the audit logs. See section 9.1.4 for additional information about how users are granted explicit read access to the audit logs.

Similar to how IT data is handled, all audit data logs are protected from deletion or modification. There are no mechanisms to explicitly delete the audit data stored on the

file system. Even if the index references are is deleted using TOE functionality, the audit.log data will still be preserved on the hard drive of the server with the _audit index.

Also similar to how IT data is handled, the most recent stored audit records will be maintained if storage runs out. There is a rollover feature in the evaluated configuration that replaces old audit data with new audit data. If the minimum hard drive storage quota specified for the Splunk instance is met, the TOE is configured to no longer allow any new IT data to be collected and begin to delete the oldest logs. This is similar with audit data; however, along with deleting the oldest audit logs, the TOE will still allow auditable events to be performed and therefore will allow audit events to be generated until all data storage is exhausted. The intention of this exception is so that an authorized user can either free the data manually using the file system of the OS upon which the search-head is installed, or increase the amount of storage that the TOE is able to use on that particular machine. When the TOE is in this state of inactivity due to low storage, users on Splunk Web will see a banner stating that storage has met or exceeded the defined threshold.

9.1.3 Cryptographic Support

The TOE utilizes OpenSSL 0.9.8n to generate cryptographic keys. OpenSSL can be utilized to encrypt and decrypt data within Splunk using many different algorithms and key sizes. However, the RSA algorithm is utilized in the evaluated configuration. The key size that is utilized in the evaluated configuration is 1024-bit keys. The TOE utilizes the RSA algorithm to encrypt generated AES keys. The AES keys encrypt the data in transit. Splunk also uses digest (hashing) algorithms to verify the integrity of Splunk data upon arrival. The digest algorithm used in the evaluated configuration for data in transit is SHA1. The SHA256 digest algorithm is used in the evaluated configuration to hash data that is being stored upon indexer machines.

Splunk generates and utilizes several .pem files for encryption and decryption, as follows:

- server.pem: This contains a X.509 certificate, the private key for the Splunk instance, and the certificate for the root CA that signed the X.509 certificate for this Splunk instance. As per the evaluated configuration, server.pem is set to use 1024-bit RSA keys and SHA1 (for data in transit).
- ca.pem: This contains a self-signed X.509 certificate and the private key for the root CA.
- cacert.pem: This contains only the X.509 certificate for the root CA.

The TOE will overwrite old keys whenever a new key is generated. This is done by leveraging standard OpenSSL functionality. In addition, organizations utilize their own certificates to protect their deployment of Splunk. However, in the evaluated configuration, only Splunk certificates are be used.

All sensitive interfaces are protected utilizing these encryption standards. 1024-bit RSA keys and SHA1 hashing is utilized to encrypt, decrypt, and sign all TSF data transferred

over a network. SSL is utilized for the user interfaces. An SSL connection exists between both user interface subsystems and the Splunkd search-head process. SSL is utilized between Splunkd processes in any deployment over a network, including but not limited to forwarder to indexer, indexer to search-head, and deployment server to deployment client. The only sensitive environment connection is the LDAP server to Splunkd for LDAP authentication, which is also protected by SSL.

9.1.4 User Data Protection

The TOE utilizes an RBAC Policy, which utilizes roles. A high level overview of the RBAC process is that users are assigned a role, and a role is assigned multiple capabilities and indexes. Access Control Lists (ACLs) are defined for Splunk objects, the definitive set of objects with ACLs is delineated in table 7-3. ACLs set which roles have what degree of access (no access, read access, read/write access). In addition, Table 9-2 below shows all of the Splunk objects within the evaluation.

Search Jobs	Commands (external python commands)
Conf-inputs	Conf-wmi
Conf-times	Eventtypes
Fieldaliases	Lookup-table-files
Macros	Manager
Nav	Props-extract
Props-lookup	Savedsearch
Scheduledviews	Sourcetype-rename
Tags	Transforms-extract
Transforms-lookup	Workflow-actions
Views	Viewstates

Table 9-2: Capabilities Within the TOE

A role in Splunk is essentially a named collection of capabilities and assignment to indexes. A capability is an action that can be performed using the Splunk interfaces. Most actions in Splunk are tied to capabilities. The complete mapping of actions to what capabilities they map to is specified in Table 7-3. Each capability represents a Boolean value that is either allow or deny. The following table shows and describes all of the capabilities within the TOE:

Capability	Description
Administer All Objects	A role with this capability has access to objects in the system. This bypasses any ACL restrictions (though not capability restrictions).
Change Authentication	Required to change authentication settings through the various authentication endpoints. Also controls whether authentication can be reloaded.

Change Own Password	Self explanatory. Some auth systems prefer to have passwords be immutable for some users.
Delete by Keyword	Required to use the 'delete' search operator. Note that this does not actually delete the data, it merely masks the data (via the index) from showing up in search results.
Edit Deployment Client	Self explanatory. The deployment client admin endpoint requires this capability for edit.
Edit Deployment Server	Self explanatory. The deployment server admin endpoint requires this capability for edit.
Edit Distribution Peer	Required to add and edit peers for distributed search.
Edit Forwarders	Required to edit settings for forwarding data. Used by TCP and Syslog output admin handlers. Includes settings for SSL.
Edit HTTP Authorizations	Required to edit and end user sessions through the httpauth-tokens endpoint.
Edit Input Defaults	Required to change the default hostname for input data in the server settings endpoint.
Edit Monitor	Required to add inputs and edit settings for monitoring files. Used by the standard inputs endpoint as well as the one-shot input endpoint.
Edit Roles	Required to edit roles as well as change the mappings from users to roles. Used by both the users and roles endpoint.
Edit Scripted	Required to create and edit scripted inputs.
Edit Search Server	Required to edit general distributed search settings like timeouts, heartbeats, and blacklists.
Edit Server	Required to edit general server settings such as the server name, log levels, etc.
Edit Splunk TCP	Required to change settings for receiving TCP input from another Splunk instance.
Edit Splunk TCP SSL	Required to list or edit any SSL specific settings for Splunk TCP input.
Edit TCP	Required to change settings for receiving general TCP inputs.
Edit UDP	Required to change settings for UDP inputs.
Edit User	Required to create, edit, or remove users. Note that Splunk users may edit certain aspects of their information without this capability. Also required to manage certificates for distributed search.
Edit Web Settings	Required to change the settings for web.conf through the system settings endpoint.
Edit Windows Admon	Required to edit the configuration for the collection of Windows Admon data.
Edit Windows Event Logs	Required to edit the configuration for the collection of Windows event logs.

Edit Windows Regmon	Required to edit the configuration for the collection of Windows registry changes.
Edit Windows WMIWGL	Required to edit the configuration for the collection of Windows WMI data.
Get Metadata	Required to use the 'metadata' search processor.
Get Typeahead	Required for typeahead. This includes the typeahead endpoint and the 'typeahead' search
Indexes Edit	Required to change any index settings like file size and memory limits.
License Tab	Required to access and change the license.
List Forwarders	Required to show settings for forwarding data. Used by TCP and Syslog output admin handlers.
List HTTP Authorizations	Required to list user sessions through the httpauth-tokens endpoint.
List Inputs	Required to view the list of various inputs. This includes input from files, TCP, UDP, Scripts, etc.
List Windows Local Available Logs	Required to list the available local logs on a Windows machine.
Request Remote Token	Required to get a remote authentication token. Used for distributing search to old 4.0.x Splunk instances. Also used for some distributed peer management and bundle replication.
Rest Apps Management	Required to edit settings for entries and categories in the python remote apps handler.
Rest Apps View	Required to list various properties in the python remote apps handler.
Rest Properties Get	Required to get information from the services/properties endpoint.
Rest Properties Set	Required to edit the services/properties endpoint.
Restart Splunkd	Required to restart Splunk through the server control handler.
Real Time Search	Required to run a realtime search.
Schedule Search	Required to schedule saved searches.
Search	Required to run a search.
Use File Operator	Required to use the 'file' search operator.

Table 9-3: Capabilities Within the TOE

When an authorized user (any user with a role that has the "Edit Role" capability) creates a role, he or she picks and chooses the capabilities that said role is allowed to perform from a master list. The default roles in Splunk are Admin, Power, User, and can_delete. The following table shows what capabilities are assigned to each role by default:

Roles Applicable	Capabilities Assigned
Power, Admin, User	Change Own Password
	Get Metadata
	Get Typeahead
	List Inputs
	Request Remote Token
	Rest Apps View
	Rest Properties Get
	Rest Properties Set
	Search
Power and Admin	Real Time Search
	Schedule Search
Admin	Admin All Objects
	Change Authentication
	Edit Deployment Client
	Edit Deployment Server
	Edit Distribution Peer
	Edit Forwarders
	Edit HTTP Authorizations
	Edit Input Defaults
	Edit Monitor
	Edit Roles
	Edit Scripted
	Edit Search Server
	Edit Server
	Edit Splunk TCP
	Edit Splunk TCP SSL
	Edit TCP
	Edit UDP
	Edit User
	Edit Web Settings

	Edit Windows Admon
	Edit Windows Event Logs
	Edit Windows Regmon
	Edit Windows WMIWGL
	Indexes Edit
	License Tab
	List Forwarders
	List HTTP Authorizations
	List Windows Local Available Logs
	Rest Apps Management
	Restart Splunkd
	Use File Operator
Can_delete	Delete by Keyword

Table 9-4: Capabilities Within the TOE

Note that can_delete contains a single exclusive capability. The typical usage of the can_delete role is to add the can_delete role to a user in addition to their normal role (admin, power, user) to grant that user the ability to delete by keyword. Note that delete by keyword doesn't actually delete information, but hides it from being returned in a search.

After picking capabilities, the authorized user is then presented a list of indexes, which is the complete list of indexes configured on the system. The indexes that ship with Splunk are as follows:

- main: The default Splunk index. All processed data is stored here by default.
- _internal: This index contains internal logs and metric from Splunk's processors.
- sampledata: This index contains some sample data for training purposes.
- _audit: This index contains all of the audit logs generated by the TOE. See Section 9.1.2 for more information on what audit information is captured.

Only the main and _audit indexes are relevant to the evaluated configuration. In addition to these preconfigured indexes, users can create additional indexes based upon their scalability, security, or organizational needs.

Similar to selecting the capabilities for a role, selecting an index simply allows that role to read the data of that index when searching, or creating saved searches or scheduled searches. There are no further limitations or specifications that can be made past determining indexes, or any sub-index functionality. When determining permissions,

capabilities and indexes are separate. There is no granularity with a capability being assigned to a role for only a specific index. However, users can be assigned multiple roles. The structure when combining multiple roles with multiple sets of capabilities and indexes is formed by determining the most permissive set of capabilities and indexes allowed. As an example, if a user has three roles, and only one role has the ability to search, then the user has the ability to search. In addition, the user would also be able to search in all indexes the user has scope over, via the roles the user is assigned.

Additionally, Splunk objects created by users have ACLs. These ACLs define which roles have what access for specific objects in the system. When changing the ACL for a specific object, the owner is presented with a list of every role currently defined in the TOE. The owner then makes a determination for each of these roles of the following options: No access, Read access, Read and write access. The ACL determination defined by the owner then gives the users with the defined roles the appropriate level of access to that specific object in the system. Each object in the system has an ACL. The only exception to this rule is that there is a capability called "Admin All Objects," which allows any role with that capability full read/write access to all objects in the system, regardless of the ACL defined. This does not mean the user can perform any function in the system; all capability and index restrictions still apply.

9.1.5 Identification and Authentication

Splunk provides a user account to every user. User accounts are created using the TOE functionality. A user account has the following attributes defined for it: username, password, and role(s). A role is defined as a collection of capabilities and indexes that that role is allowed to utilize. The default roles available in a Splunk installation are as follows: Admin, Power, User and can_delete. These roles are described in more detail in Section 9.1.4.

Users are generated by any user that belongs to a role with the "Edit User" capability. A default Administrator account is provided with the initial installation of the TOE with the username and password combination "admin/changeme." This Administrator account has full scope over all of the capabilities and indexes of the TOE.

Before utilizing the TOE's functionality, all users must both successfully identify and authenticate themselves by providing their correct username and password when utilizing Splunk Web or the Splunk CLI. There are no actions that a user can perform using the TOE functionality without first being authenticated through these interfaces.

There are two authentication mechanisms utilized in the TOE: Splunk authentication and LDAP authentication. Authorized users are able to select the authentication method used within the configuration options of the TOE. Splunk authentication simply stores the user information locally and checks username and password combinations against this information whenever a user attempts to log in.

The LDAP authentication is utilized as follows:

- A user attempts to log in by entering username and password combination via Splunk Web or Splunk CLI.
- The TOE sends an LDAP search with the username inputted to determine if the username is in the LDAP user table.
- The LDAP store returns the username, distinguished name, and attribute used for mapping the user to an LDAP group to the TOE.
- The TOE then sends a request to the LDAP store to see if a returned user attribute is in a group.
- The LDAP server returns the group to the TOE.
- The TOE sends the distinguished name and password to the LDAP store via a bind using OpenLDAP over SSL, and will ultimately receive a determination on whether or not the user has successfully authenticated.

Any LDAP query made by the TOE before successful authentication that produces an error or returns no results will return the error to the TOE. Subsequently, the TOE will notify the user that authentication has failed. Groups, in terms of the LDAP store, refer to roles in the TOE.

Upon logging into the TOE, a user's attributes and roles are associated with a session object. A session is created by generating a random number, which is the primary session identifier, and associating this unique number with the user attributes within the TOE. The user-subject binding will also keep track of the last activity under the session. Whenever a user attempts to perform an action, the TOE checks authorizations by comparing the action that is attempted with the role information that is mapped to the user's session.

Any changes made to a user during a session, including the roles it is assigned, will update the session object for that specific user. Because every action a user performs requires a check for authorization, the TOE performs secure revocation on an architectural level. If a user's role is edited to remove capabilities or indexes, that change will be reflected upon the next attempted user action since the session object is updated with new information. Similarly, if a user is deleted the session is also deleted, and the next time a user tries to perform any action, that user will be brought back to the login prompt.

Based upon a configurable time threshold, the TOE will also continuously check for any sessions that have been inactive for a time period reaching or exceeding the defined threshold. When the difference between current time and the time of the last activity within a session becomes greater than the configurable threshold, the session in question is also removed. The user within that session will be brought back to the login prompt once an action is attempted.

9.1.6 Security Management

The security management of the TOE is controlled by user actions that are performed under the RBAC policy as described in Section 9.1.4.

All functions within the system operate upon the following conditions: roles, capabilities assigned to roles, indexes assigned to roles, ACLs assigned to objects, and roles assigned to users. Capabilities impact the scope of the TOE functionality itself that a role is allowed to perform. Indexes impact the scope of TSF data that a role is allowed to access. ACLs impact the scope of TSF objects that a role is allowed to access. Depending on the function, object, and index utilized within a management function of the TOE, different access restrictions may be required. This is detailed within Table 7-3. A summary of the actions users can perform in the TOE are as follows: configuring the TOE, managing users, managing roles, searching IT/audit data, and managing TOE objects.

In conjunction with the overarching management capabilities of the TOE with respect to the RBAC policy utilized, the TOE also provides mechanisms to maintain and configure the security attributes within the TOE. Table 7-4 details the actions available to each security attribute and also provides the authorization (security attribute) required to perform the action.

The TOE provides restrictive default values for these security attributes. When a user is initially being created by an authorized user, the default role is "user," which is the role with least privileges in the system. The authorized user creating the new user must then explicitly grant privileges to the newly-created user. The default attributes of the default roles are described in Table 9-4.

The primary security attributes within the system when determining what actions are allowed are roles. The default roles in the system are the following: admin, power, user, and can_delete. Users with the Edit Role functionality (as shown in Table 7-3) can edit these default roles, create and modify new roles, or delete roles. As talked about in Section 9.1.4, when creating a role an authorized user must specify a name for the role, a selection of applicable capabilities for the role to embody, and a selection of applicable indexes for the role to be granted access. As stated above, a user must be assigned to one or more roles before the user can perform any TSF-related action on the TOE.

A user that possesses the Edit User capability can utilize the TOE functionality to edit users. In doing so, this authorized user can revoke the user account of a user, or just the role of the user. Any changes of the roles a user possesses either in the editing of the user or the editing of the role will update the session object generated upon authentication upon the next user action. In addition, whenever a user is deleted or the user loses all roles through the usage of these methods, the session object will be terminated. This is still performed through the same updating method as described above, but the session will become invalid if these conditions occur.

9.1.7 Protection of the TSF

The TOE utilizes OpenSSL 0.9.8n functionality to prevent unauthorized disclosure of data and detect modification of TSF data sent to other trusted IT products. As shown in a previous section, the TOE utilizes OpenSSL to encrypt data in transmission with 1024-bit RSA keys (generated using AES algorithm) to prevent against unauthorized disclosure. The TOE also uses the SHA1 hashing algorithm to provide a hash signature for all data sent to external products. This hash signature is utilized to verify if the data was altered in any way during transmission. When utilizing the SHA1 hashing in the TOE, the TOE will drop any packets that fail verification tests upon arrival. The only external IT product that requires secure data transport is the connection to the LDAP server for LDAP authentication.

The TOE uses these same technologies and algorithms (OpenSSL, 1024-bit RSA, SHA1) to protect the data moving between TOE components. This includes all data transferred between multiple Splunkd components over the network. These connections include, but are not limited to: forwarder to indexer, indexer to search-head, and deployment server to deployment client. When using SSL to create this trusted path between TOE components, the TOE provides mutual authentication over SSL in addition to encryption of data. The TOE contains the ability to authorize CAs for this purpose.

The TOE also uses these methods to create a trusted path between users and the TOE. OpenSSL is utilized to secure all data transferred between Splunk Web or Splunk CLI to the Splunkd search-head. This is technically a TOE to TOE connection, but connections to Splunk Web and Splunk CLI are local, while all the encryption and hashing occurs between these user interface subsystems and the Splunkd search-head subsystem utilized. Remote users initiate this trusted path by either launching the Splunk CLI process or connecting to Splunk Web on their end. This trusted path is required for initial user authentication, as well as any and all transfers of data or management actions initiated by the user.

9.1.8 High Availability

The TOE provides mechanisms for high availability. Forwarders that are configured for auto-load balancing between multiple indexers will attempt to send data to the inactive indexer, receive no response from the indexer, and then switch their connection to one of the other indexers to which it is configured to forward. None of the data being sent to an inactive indexer is lost or skipped over when this occurs; the data attempted to be sent to the inactive indexer will be sent to one of the active indexers in the deployment.

Another mechanism for high availability is data cloning. Indexers within the Splunk system can be configured to clone all data being indexed. This cloned data will then be forwarded to another indexer. This allows for additional assurance that indexed data will be available for searching if one or more specific indexers goes down. However, this requires additional TOE configuration outside of the initial TOE configuration by an authorized user to utilize this specific indexer for indexing and searching. This

functionality provides data reliability search availability in the event of a single indexer failure.

9.1.9 Security Architecture

Splunk has multiple architecture features that enhance the security functions of the TOE.

Splunkd runs on separate threads/processes in specific instances. The most important instance is whenever a Splunkd process is created upon the initiation of a search. This Splunkd process operates solely to perform that search and utilizes no other Splunkd functionality. Each of the Splunkd processes has its own memory space on the machine upon which it is running. For the example of separate Splunkd search processes, this implies that whenever a Splunkd search process breaks for any reason, be it timing out, using too many resources, or any other fail state, the process can be terminated without affecting the Splunk deployment as a whole or on that specific machine.

Whenever a user initiates a search, in addition to the search request itself, the session token for that specific user is sent. Once the search is processed, the token is sent back to the user along with the results for the specific search. The session tokens are utilized the same way in both the Splunk Web and Splunk CLI processes. This allows both the TOE and the user to verify that the data sent through these transactions are authentic through the use of session tokens. Session files are generated for each user. The session file contains the keys utilized within the session.

All user actions are default deny; the authorization mechanisms within the TOE are required to be active before any action takes place. This means users cannot formulate and send commands to be processed to the TOE and bypass the authorization mechanisms upon initial boot or any other failure state of the authentication mechanism. This protects against users attempting to perform commands outside of the scope of their role.

9.2 TOE Summary Specification Rationale

This section identifies the security functions provided by the TOE mapped to the security functional requirement components contained in this ST. This mapping is provided in the following table.

Security Function	Security Functional Components
IT Data Indexing (FAU)	FAU_GEN_EXT.1 IT Data Collection
	FAU_SAR_EXT.1 IT Data Review
	FAU_SAR_EXT.2 Restricted IT data review
	FAU_SAR_EXT.3 Selectable IT Data Review
	FAU_STG_EXT.2 Guarantees of IT Data Availability
	FAU_STG_EXT.3 Action in Case of IT Data Loss
	FAU_STG_EXT.4 Prevention of IT Data Loss
Security Audit (FAU)	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR.2 Restricted audit review
	FAU_SAR.3 Selectable audit review
	FAU_STG.2 Guarantees of audit data availability
	FAU_STG.3 Action in case of audit data loss
Cryptographic Support (FCS)	FCS_CKM.1 Cryptographic key generation
	FCS_CKM.4 Cryptographic key destruction
	FCS_COP.1 Cryptographic operation
User Data Protection (FDP)	FDP_ACC.1 Access control policy
	FDP_ACF.1 Access control functions
Identification and Authentication (FIA)	FIA_ATD.1 User attribute definition
	FIA_UAU.2 User authentication before any action
	FIA_UAU.5 Multiple authentication mechanisms
	FIA_UAU.6 Re-authenticating
	FIA_UID.2 User identification before any action
	FIA_USB.1 User-subject binding
Security Management (FMT)	FMT_MOF.1 Management of security functions behavior
	FMT_MSA.1 Management of security attributes
	FMT_MSA.3 Static attribute initialization
	FMT_MTD.1 Management of TSF data
	FMT_REV.1 Revocation
	FMT_SMF.1 Specification of Management Functions
	FMT_SMR.1 Security roles
Protection of the TSF (FPT)	FPT_FLS.1 Failure with preservation of secure state
	FPT_ITC.1 Confidentiality of exported TSF data
	FPT_ITI.1 Integrity of exported TSF data
	FPT_ITT.1 Basic internal TSF data transfer protection
Resource Utilization (FRU)	FRU_FLT.1 Fault tolerance
Trusted Path/Channels (FTP)	FTP_TRP.1 Trusted path

Table 9-5: Security Functional Components for the TOE

9.2.1 IT Data Indexing

This section maps directly to the information found in Section 9.1.1. This security classification addresses the following requirements: FAU_GEN_EXT.1, FAU_SAR_EXT.1, FAU_SAR_EXT.2, FAU_SAR_EXT.3, FAU_STG_EXT.2, FAU_STG_EXT.3, FAU_STG_EXT.4.

The referred section specifically addresses FAU_GEN_EXT.1 by providing a list of what kind of data the TOE can collect, and providing a list of what data is collected. FAU_SAR_EXT.1, FAU_SAR_EXT.2, and FAU_SAR_EXT.3 are satisfied by providing information on the search functionality, which is the primary way to review IT data, and how users are authorized to view said data. FAU_STG_EXT.2, FAU_STG_EXT.3, and FAU_STG_EXT.4 are satisfied by describing how data is stored by supplying information about indexes. It also explains that data is rolled over based on size of the data trail and age of the data, and that data cannot be modified or deleted by users according to lack of TOE functionality to perform those actions. Information about setting quotas also shows how data storage is exhausted.

9.2.2 Security Audit

This section maps directly to the information found in Section 9.1.2. This security classification addresses the following requirements: FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_STG.2, FAU_STG.3, FAU_STG.4.

FAU_GEN.1 and FAU_GEN.2 are addressed by showing which actions are audited (all user actions, login, startup/shutdown) and providing all the data that is collected in an audit log, including the user who performed an action. FAU_SAR.1 and FAU_SAR.2 are demonstrated by showing that only users with access to the _audit index can view audit data. FAU_SAR.3 is explained by detailing the search functionality and how it pertains to viewing audit records. FAU_STG.2, FAU_STG.3, and FAU_STG.4 are addressed by showing that the TOE indexing functionality is interrupted when storage is full, and actions can be performed while the TOE frees up space by rolling over old data.

9.2.3 Cryptographic Support

This section maps directly to the information found in Section 9.1.3. This security classification addresses the following requirements: FCS_CKM.1, FCS_CKM.4, FCS_COP.1.

FCS_CKM.1 and FCS_CKM.4 are addressed by showing that the TOE uses OpenSSL with 1024-bit RSA keys and SHA1 signatures to generate and destroy keys. FCS_COP.1 is addressed by detailing the interfaces that utilize cryptography within the TOE.

9.2.4 User Data Protection

This section maps directly to the information found in Section 9.1.4. This security classification addresses the following requirements: FDP_ACC.1, FDP_ACF.1.

FDP_ACC.1 and FDP_ACF.1 are addressed by explaining how users, roles, capabilities, allowed indexes, and ACLs all interact. This explanation shows the complex RBAC policy utilized by the TOE.

9.2.5 Identification and Authentication

This section maps directly to the information found in Section 9.1.5. This security classification addresses the following requirements: FIA_ATD.1, FIA_UAU.2, FIA_UAU.5, FIA_UAU.6, FIA_UID.2, FIA_USB.1.

FIA_ATD.1 is addressed by showing the attributes a user has: username, password, and roles. FIA_UAU.2 and FIA_UID.2 are detailed by explaining the user interfaces in the TOE and how nothing can be done until they successfully log in to the TOE. FIA_UAU.5 is addressed by showing that the TOE can use native authentication or LDAP authentication as well as by showing how LDAP is utilized. FIA_UAU.6 is explained by detailing how users are disconnected due to inactivity. FIA_USB.1 is addressed by explaining how users are bound to their roles by utilizing a session object, and how session objects can be removed due to the alteration of the user or role in the system.

9.2.6 Security Management

This section maps directly to the information found in Section 9.1.6. This security classification addresses the following requirements: FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_REV.1, FMT_SMF.1, FMT_SMR.1.

FMT_MOF.1, FMT_MTD.1, and FMT_SMF.1 are addressed by referring to Table 7-3, which details all of the functions available within the system based upon capability, index, and ACL. This upholds the RBAC policy and provides the object, object, and attribute mapping. FMT_MSA.1 is explained by referring to Table 7-4, which details all of the security attributes involved in the RBAC policy and how each is accessed, edited, and/or assigned. FMT_MSA.3 is shown by stating that users are created with the user role by default, giving them limited access to the TSF. It is also shown by describing the methods users can perform to alter default attribute values. FMT_REV.1 is addressed by stating that users can have their user account or roles revoked at any time by authorized users. In addition, this section states that the session object is updated whenever a user's role is changed, which allows for real-time changes to roles or users that have somewhat immediate impact (sessions are only updated when an action is attempted, which is adequate for protecting the TSF). Using these mechanisms a user can terminate a session object by making it invalid by removing the user or removing all roles from a user. FMT_SMR.1 is explained by showing the default roles in the system and detailing the role editing functionality. It also explains that users cannot perform any TSF-related actions before having a role mapped to them.

9.2.7 Protection of the TSF

This section maps directly to the information found in Section 9.1.7. This security classification addresses the following requirements: FPT_ITC.1, FPT_ITI.1, FPT_ITT.1, FTP_TRP.1.

FPT_ITC.1 and FPT_ITI.1 are addressed by showing the connection to the external LDAP server that is encrypted with OpenSSL. FPT_ITT.1 is explained by detailing the TOE components that have encryption in between them in the deployment. FTP_TRP.1 is addressed by showing the user connection to the user interfaces is encrypted and hashed as well.

9.2.8 High Availability

This section maps directly to the information found in Section 9.1.8. This security classification addresses the following requirements: FPT_FLS.1, FRU_FLT.1.

FPT_FLS.1 and FRU_FLT.1 are explained by showing that there are multiple functions to protect the TOE in the event of a single TOE component failing. The following mechanisms were described: auto-load balancing, data cloning, and separate search processes.

10 Security Problem Definition Rationale

10.1 Security Objectives Rationale

The following table provides a mapping with rationale to identify the security objectives that address the stated assumptions and threats.

Assumption	Objective	Rationale
A.ADMIN One or more users authorized by the Operational Environment will be assigned to install, configure and manage the TOE and the security of the information it contains.	OE.ADMIN One or more authorized users will be assigned to configure the Operational Environment, and install, configure, and manage the TOE and the security of the information it contains.	OE.ADMIN maps to A. ADMIN in order to ensure that only the users authorized by the TOE will install and configure the TOE to bring it into the evaluated configuration. During operation only the users authorized by the TOE will be able to manage the TOE in a manner that maintains its ADMIN objectives.
A.NOEVIL Users of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.	OE.NOEVIL All users of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.	OE.NOEVIL directly maps to A.NOEVIL and ensures that all users of the TOE are properly trained in the configuration and usage of the TOE and will follow the guidance provided.
A.PATCHES Users exercise due diligence to update the TOE with the latest patches and patch the Operational Environment (e.g., OS and database) so they are not susceptible to network attacks.	OE.ADMIN One or more authorized users will be assigned to configure the Operational Environment, and install, configure, and manage the TOE and the security of the information it contains.	OE.ADMIN maps to A. PATCHES in order to ensure that the users authorized by the TOE will properly patch the TOE and the Operational Environment in a manner that maintains their security objectives.
A.FILESYS The security features offered by the Operational Environment protect the files used by the TOE.	OE.FILESYS The security features offered by the Operational Environment will protect the files used by the TOE.	OE.FILESYS maps to A.FILESYS in order to ensure that the Operational Environment's native security features are utilized when securing data relevant to the TOE outside of its boundary.
A.SECURE The network which the TOE is monitoring is expected to be secure to protect the transfer of data from remote data sources and email messages sent to the SMTP Server.	OE.SECURE The network which the TOE is monitoring will be secured to protect data being sent to the TOE as well as SMTP messages sent from the TOE.	OE.SECURE maps to A.SECURE in order to ensure that the data being transferred to the TOE via the monitored network as well as the SMTP traffic generated by the TOE is properly secured such that the TOE receives accurate data and TOE alerts are not disclosed to

		unauthorized users.
A.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.	OE.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.	OE.LOCATE maps to A.LOCATE in order to ensure that physical security is provided in the environment where the TOE operates.

Table 10-1: Assumption to Objective Mapping

Threat	Objective	Rationale
T.ACCESS Authorized users could gain electronic access to protected TOE resources by attempting to establish a connection that they are not permitted to perform.	O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	O.ACCESS addresses T.ACCESS by providing the authorized administrators of the TOE with the capability to specify access restrictions on the protected TOE resources to authenticated TOE users.
	O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.	O.MANAGE addresses T.ACCESS by ensuring only users authorized by the TOE can use the TOE provided resources to manage and monitor the TOE's auditing capabilities.
	O.SELF_PROTECTION The TOE will preserve a secure state even in the presence of adversarial activity and ensure overall TOE functionality remains operational when a component of the TOE fails.	O.SELF_PROTECTION addresses T.ACCESS by ensuring connectivity of failed components are reinitialized prior to resources being accessed.
T.ADMIN_ERROR A user may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.	O.ROBUST_ADMIN_GUIDANCE The TOE will provide the TOE's users with the necessary information for secure delivery, installation, management, and operation of the TOE.	O.ROBUST_ADMIN_GUIDANCE helps to mitigate T.CONFIG_ERROR by ensuring the TOE users have guidance that instructs them how to administer the TOE in a secure manner and to provide the TOE users with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that a user might make that could cause the TOE to be configured in a way that is unsecure.
	O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.	O.MANAGE addresses T.ADMIN_ERROR by ensuring only users authorized by the TOE can use the TOE provided resources to manage and monitor the TOE's auditing capabilities.

Threat	Objective	Rationale
T.AUDIT_COMPROMISE A malicious user or process may view audit event records or alert data, cause the records or alert data to be lost or modified, or prevent future audit event records from being recorded and alert data from being sent, thus masking a user's action.	O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE, the information it collects, and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	O.AUDIT addresses T.AUDIT_COMPROMISE by providing the users authorized by the TOE with the tools necessary to monitor users or processes to ensure that misuse of the TOE or the Operational Environment does not occur.
	O.ALERT The TOE will provide measures for determining security violations and will create alarms when audit data that represents any of these violations is processed.	O.ALERT addresses T.AUDIT_COMPROMISE by providing the users with the ability of receiving alert notifications from the TOE when events are considered to be a security violation based on defined policy.
	O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	O.ACCESS addresses T.AUDIT_COMPROMISE by providing the authorized administrators of the TOE with the capability to specify access restrictions on the protected TOE resources to authenticated TOE users.
	OE.AUDIT The Operational Environment will provide local access control, and storage of the event audit records which are stored on the machine where the TOE is installed.	OE.AUDIT helps to mitigate T.AUDIT_COMPROMISE by providing the TOE with the Operational Environment's ability to store the audit data and to protect the event audit records from local access.
	OE.SYSTIME The Operational Environment will provide reliable system time.	OE.SYSTIME helps to mitigate T.AUDIT_COMPROMISE by ensuring the accuracy of the tools necessary to monitor user activity as provided via O.AUDIT.
T.EAVESDROPPING A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data.	O.EAVESDROPPING The TOE will encrypt TSF data that traverses the network to prevent malicious users from gaining unauthorized access to TOE data.	O.EAVESDROPPING mitigates T.EAVESDROPPING by ensuring that all communication to/from the TOE are not sent unless it is encrypted by the TOE.

Threat	Objective	Rationale
T.INTERFACE_ABUSE A malicious user may attempt to communicate with the TOE's interfaces in ways that exploit flaws, subvert the TOE, or defeat the operation of its security mechanisms.	O.SELF_PROTECTION The TOE will preserve a secure state even in the presence of adversarial activity and ensure overall TOE functionality remains operational when a component of the TOE fails.	O.SELF_PROTECTION addresses T.INTERFACE_ABUSE by asserting that all interfaces will perform properly and securely in the event of an attack using one of the external interfaces.
T.MASK Users whether they be malicious or non-malicious, could gain unauthorized access to the TOE by bypassing identification and authentication countermeasures.	O.AUTH The TOE will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the TOE.	O.AUTH addresses T.MASK by ensuring that all users that try and access the TOE become identified and authenticated before access is granted.
T.MISUSE Users of an IT product the TOE monitors may perform undesirable actions using the IT product in question, whether by performing malicious actions upon the product, by utilizing functions within the IT product that adversely affect the system it interacts with, or by altering the configuration to be insecure.	O.INDEX The TOE will provide measures for collecting security relevant events from configured IT products. These events that will assist the authorized users in detecting misuse of the IT products, the information they collect, and/or their security features that would compromise the integrity of the IT product and violate the security objectives of the IT product.	O.INDEX addresses T.MISUSE by providing the users authorized to utilize the TOE with the capability to perform monitoring functions by utilizing the data that the TOE collects. This data is potentially able to allow these users to find undesirable actions performed upon the IT products the TOE monitors.
T.STEALTH A malicious user or process could perform suspicious activities against the TOE or objects in the Operational Environment monitored by the TOE without a TOE user authorized by the Operational Environment becoming aware of this behavior.	O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE, the information it collects, and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	O.AUDIT addresses T.MASK by providing the users authorized by the TOE with the tools necessary to monitor users or processes to ensure that misuse of the TOE or the Operational Environment does not occur.
	O.ALERT The TOE will provide measures for determining security violations and will create alarms when audit data that represents any of these violations is processed.	O.ALERT addresses T.STEALTH by providing the users with the ability of receiving alert notifications from the TOE when events are considered to be a security violation based on defined policy.
	O.AUTH The TOE will provide measures to	O.AUTH addresses T.STEALTH by ensuring that

Threat	Objective	Rationale
	uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the TOE.	all users that try and access the TOE become identified and authenticated before access is granted.
	O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	O.ACCESS helps to mitigate T.STEALTH by providing the TOE with access control functions, which restricts access to the TOE and its objects to users which have been authorized access by an administrator.
	OE.AUDIT The Operational Environment will provide local access control, and storage of the event audit records which are stored on the machine where the TOE is installed.	OE.AUDIT helps to mitigate T.STEALTH by providing the TOE with Operational Environment's ability to store the audit data and protect the event audit records from local access.
	OE.SYSTIME The Operational Environment will provide reliable system time.	OE.SYSTIME helps to mitigate T.STEALTH by ensuring the accuracy of the tools necessary to monitor user activity as provided via O.AUDIT.

Table 10-2: Threat to Objective Mapping

10.2 Operational Security Policy Rationale

There are no Organizational Security Policies that apply to the TOE.

10.3 Security Functional Requirements Rationale

The following table provides a mapping with rationale to identify the security functional requirement components that address the stated TOE objectives.

Objective	Security Functional Components	Rationale
O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the	FDP_ACC.1 Access control policy	FDP_ACC.1 details that the policy defined by FDP_ACF.1 is enforced on all users looking to perform any action on the TOE.
	FDP_ACF.1 Access control functions	FDP_ACF.1 details all of the access control factors: capabilities, indexes, roles, and ACLs. It also shows some use cases for specific scenarios of the RBAC system.

Objective	Security Functional Components	Rationale
TOE.	FIA_ATD.1 User attribute definition	FIA_ATD.1 shows all of the user attributes that are utilized to authorize users, including user role.
	FIA_USB.1 User-subject binding	FIA_USB.1 shows how users and their roles are associated in a session object. It also details how a user loses his or her session object based upon administrative changes.
	FMT_REV.1 Revocation	FMT_REV.1 details how a user loses his or her session object based upon administrative changes.
	FMT_SMR.1 Security roles	FMT_SMR.1 defines that there are roles in the system and that users are associated with their role for making authorization decisions.
O.ALERT The TOE will provide measures for determining security alerts when audit data or IT records that represent any of these alerts is recorded.	FAU_STG_EXT.4 Prevention of IT Data Loss	FAU_STG_EXT.4 states that the TOE will display an alert once IT data storage is exhausted.
	FAU_STG.4 Prevention of audit data loss	FAU_STG.4 states that the TOE will display an alert once audit storage is exhausted.
O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE, the information it collects, and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	FAU_GEN.1 Audit data generation	FAU_GEN.1 defines the behavior of the TSF which causes security relevant events to be generated and enumerates the data which is contained within these events.
	FAU_GEN.2 User identity association	FAU_GEN.2 confirms that all relevant auditable events include subject identity for the purposes of accountability.
	FAU_SAR.1 Audit review	FAU_SAR.1 provides the ability for all authorized users to read audit data using the search functionality.
	FAU_SAR.2 Restricted audit review	FAU_SAR.2 states that only authorized users can read audit data. Authorized users are any users with the search capability and _audit index mapped to their role.
	FAU_SAR.3 Selectable audit review	FAU_SAR.3 defines the ability of the TOE to selectively display audit data based on event type and either a time

Objective	Security Functional Components	Rationale
		range or number of events.
	FAU_STG.2 Guarantees of audit data availability	FAU_STG.2 states that no TOE functionality exists to modify or delete audit records and that old logs will not be deleted once storage is exhausted.
	FAU_STG.3 Action in case of possible data loss	FAU_STG.3 defines the ability of the TOE to automatically prune the oldest audit records in the database when a configurable number of records has been exceeded.
	FAU_STG.4 Prevention of audit data loss	FAU_STG.4 describes the process the TOE takes to assure that space is available for the newest audit logs, by deleting the oldest audit logs.
O.AUTH The TOE will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the TOE.	FIA_ATD.1 User attribute definition	FIA_ATD.1 defines the security-relevant attributes of all users. This includes attributes related to authentication.
	FIA_UAU.2 User authentication before any action	FIA_UAU.2 requires users to authenticate to the TOE before any TSF-mediated actions are allowed.
	FIA_UAU.5 Multiple authentication mechanisms	FIA_UAU.5 defines the two authentication mechanisms in the TOE: Splunk authentication and LDAP authentication.
	FIA_UAU.6 Re-authentication	FIA_UAU.6 states that users must re-authenticate upon session inactivity.
	FIA_UID.2 User identification before any action	FIA_UID.2 requires users to identify themselves to the TOE before any TSF-mediated actions are allowed.
	FIA_USB.1 User-subject binding	FIA_USB.1 defines the mapping between users and roles and the creation of a session object.
O.EAVESDROPPING The TOE will encrypt TSF data that traverses the network to prevent malicious users from gaining unauthorized access to TOE data.	FCS_CKM.1 Cryptographic key generation	FCS_CKM.1 requires the TOE to generate proper cryptographic keys for use in encrypting sensitive data.
	FCS_CKM.4 Cryptographic key destruction	FCS_CKM.4 requires the TOE to destroy cryptographic keys used in encrypting sensitive data. Keys are destroyed when new keys are generated.
	FCS_COP.1 Cryptographic	FCS_COP.1 requires the TOE to utilize the generated cryptographic

Objective	Security Functional Components	Rationale
	operation	keys in protecting all data transferred to and from users, other TOE components, and external IT products.
	FPT_ITC.1 Confidentiality of exported TSF data	FPT_ITC.1 requires the TOE to make all security-relevant data sent to and from external products protected against disclosure. This is done by creating a trusted channel with the encryption mechanisms described in FCS_COP.1.
	FPT_ITI.1 Integrity of exported TSF data	FPT_ITI.1 requires the TOE to make all security-relevant data sent to and from external products protected against modification. This is done by creating A trusted channel with the encryption mechanisms described in FCS_COP.1.
	FPT_ITT.1 Basic internal TSF data transfer protection	FPT_ITT.1 requires the TOE to make all security-relevant data sent between remote TOE components protected against modification. This is done by creating a trusted channel with the encryption mechanisms described in FCS_COP.1.
	FTP_TRP.1 Trusted path	FTP_TRP.1 requires the TOE to make all security-relevant data sent to and from users protected against modification. This is done by creating a trusted path with the encryption mechanisms described in FCS_COP.1.
O.INDEX The TOE will provide measures for collecting security relevant events from configured IT products. These events that will assist the authorized users in detecting misuse of the IT products, the information they collect, and/or their security features that would compromise the integrity of the IT product and violate the security objectives of the IT product.	FAU_GEN_EXT.1 IT Data Collection	FAU_GEN_EXT.1 details the types of IT data that the TOE collects, the types of data sources, and the minimum contents of the data that is collected. This provides assurance that sufficient data is available to TOE users for analysis.
	FAU_SAR_EXT.1 IT Data Review	FAU_SAR_EXT.1 details the methods that TOE users can view data by describing the default searchable fields in each indexed data log. This provides assurance that TOE users will be able to interpret the data appropriately.
	FAU_SAR_EXT.2 Restricted IT data review	FAU_SAR_EXT.2 states that all users except the ones explicitly allowed as defined by FAU_SAR_EXT.1 are not

Objective	Security Functional Components	Rationale
		allowed to view collected IT data.
	FAU_SAR_EXT.3 Selectable IT Data Review	FAU_SAR_EXT.3 describes the method that users view the IT data collected by the TOE, which is the search functionality.
	FAU_STG_EXT.2 Guarantees of IT Data Availability	FAU_STG_EXT.2 states that no TOE user can delete or modify data, and that all data is preserved indefinitely within the TOE functionality. This provides assurance that the indexed data is complete and accurate.
	FAU_STG_EXT.3 Action in Case of IT Data Loss	FAU_STG_EXT.3 requires the TOE to stop indexing and preserve all collected data once storage is exhausted. This provides assurance by preserving the previously collected data.
	FAU_STG_EXT.4 Prevention of IT Data Loss	FAU_STG_EXT.4 requires the TOE to send an alarm when data storage is exhausted, giving TOE users ample warning that storage is exhausted and will allow authorized users to rectify the issue.
O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.	FMT_MOF.1 Management of security functions behavior	FMT_MOF.1 defines the capabilities that can be enabled or disabled on specific roles by authorized users.
	FMT_MSA.1 Management of security attributes	FMT_MSA.1 defines the management actions and which combinations of capabilities, indexes, and ACLs allow users to perform specific functions. Authorized users are able to perform these access control policy changes.
	FMT_MSA.3 Static attribute initialization	FMT_MSA.3 requires all security-relevant attributes have secure default values, and that an authorized user has the ability to override default values.
	FMT_MTD.1 Management of TSF data	FMT_MTD.1 defines the conditions which combinations of capabilities, indexes, and ACLs allow users to perform specific actions on TSF data. Authorized users are able to perform these access control policy changes.
	FMT_SMF.1 Specification of Management Functions	FMT_SMF.1 defines the functions that can be performed by specific types of

Objective	Security Functional Components	Rationale
		users of the TOE.
O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	AGD_OPE.1 Operational User Guidance	AGD_OPE.1 describes the proper delivery, usage, and management of the TOE.
O.SELF_PROTECTION The TOE will preserve a secure state and ensure overall TOE functionality remains operational when a component of the TOE fails.	FPT_FLS.1 Failure with preservation of secure state	FPT_FLS.1 requires that the TOE shall continue to operate in a secure manner if a specific TOE component fails.
	FRU_FLT.1 Fault Tolerance	FRU_FLT.1 states that the TOE's primary functionality will continue operation in the event of a TOE component failing.

Table 10-3: Security Functional Requirements Rationale

10.4 EAL2 Justification

The threats that were chosen are consistent with an attacker of basic attack potential, therefore EAL2 augmented with ALC_FLR.1 was chosen for this ST. ALC_FLR.1 is not required, but provides additional quality assurance to the product.

10.5 Requirement Dependency Rationale

The table below lists each requirement from claimed Security Functional Requirements with a dependency and indicates whether the dependent requirement is included. If a dependency has not been met, a short rationale is provided to show why the dependency is not included.

Functional Component	Dependency	Included
FAU_GEN_EXT.1	FPT_STM.1	NO, the environment fulfills this dependency (OE.SYSTIME).
FAU_SAR_EXT.1	FAU_GEN_EXT.1	YES
FAU_SAR_EXT.2	FAU_SAR_EXT.1	YES
FAU_SAR_EXT.3	FAU_SAR_EXT.1	YES
FAU_STG_EXT.2	FAU_GEN_EXT.1	YES
FAU_STG_EXT.3	FAU_GEN_EXT.1	YES
FAU_STG_EXT.4	FAU_STG_EXT.2	YES

FAU_GEN.1	FPT_STM.1	NO, the environment fulfills this dependency (OE.SYSTIME).
FAU_SAR.1	FAU_GEN.1	YES
FAU_SAR.2	FAU_SAR.1	YES
FAU_SAR.3	FAU_SAR.1	YES
FAU_STG.2	FAU_GEN.1	YES
FAU_STG.3	FAU_GEN.1	YES
FAU_STG.4	FAU_STG.1	YES (Hierarchy: FAU_STG.2)
FCS_CKM.1	FCS_CKM.2 or FCS_COP.1	YES (FCS_COP.1)
	FCS_CKM.4	YES
FCS_CKM.4	FDP_ITC.1, FDP_ITC.2, or FCS_CKM.1	YES (FCS_CKM.1)
FCS_COP.1	FDP_ITC.1, FDP_ITC.2, or FCS_CKM.1	YES (FCS_CKM.1)
	FCS_CKM.4	YES
FDP_ACC.1	FDP_ACF.1	YES
FDP_ACF.1	FDP_ACC.1	YES
	FMT_MSA.3	YES
FIA_UAU.2	FIA_UID.1	YES (Hierarchy: FIA_UID.2)
FIA_USB.1	FIA_ATD.1	YES
FMT_MOF.1	FMT_SMR.1	YES
FMT_MSA.1	FDP_ACC.1 or FDD_ICF.1	YES (FDP_ACC.1)
	FMT_SMR.1	YES
	FMT_SMF.1	YES
FMT_MSA.3	FMT_MSA.1	YES
FMT_MTD.1	FMT_SMR.1	YES
FMT_REV.1	FMT_SMR.1	YES
FMT_SMR.1	FIA_UID.1	YES (Hierarchy: FIA_UID.2)
FRU_FLT.1	FPT_FLS.1	YES

Table 10-4: Requirement Dependencies

10.6 Assurance Measures

The SARs for this evaluation have been chosen because they are consistent with the package claim of EAL2. Augmentations to this claim include ALC_FLR.1. ALC_FLR.1

provides assurance that the TOE is updated in a well-defined manner that is consistent with the development security procedures outlined in ALC_DVS.1.

The following table identifies the SARs for this ST.

Component	Document(s)	Rationale
ADV_ARC.1 Security Architecture Description	TOE Design Specification Document for Splunk Inc. Splunk 4.1.7 version 0.2	This document describes the security architecture of the TOE.
ADV_FSP.2 Security-enforcing functional specification	Functional Specification Document for Splunk Inc. Splunk 4.1.7 version 0.2	This document describes the functional specification of the TOE with complete summary.
ADV_TDS.1 Basic Design	TOE Design Specification Document for Splunk Inc. Splunk 4.1.7 version 0.2	This document describes the architectural design of the TOE.
AGD_OPE.1 Operational User Guidance	• Splunk Admin Manual Version: 4.1.7 • Splunk Application Management Version: 4.1.7 • Splunk Developer Manual Version: 4.1.7 • Splunk Installation Manual Version: 4.1.7 • Splunk Knowledge Manager Manual Version: 4.1.7 • Splunk Release Notes Version: 4.1.7 • Splunk Search Reference Version: 4.1.7 • Splunk User Manual Version: 4.1.7	These documents describe the operational user guidance for the TOE.
AGD_PRE.1 Preparative Procedures	• Splunk Admin Manual Version: 4.1.7 • Splunk Application Management Version: 4.1.7 • Splunk Developer Manual Version: 4.1.7 • Splunk Installation Manual Version: 4.1.7	This document describes the preparative procedures that need to be done prior to installing the TOE.

Component	Document(s)	Rationale
	• Splunk Knowledge Manager Manual Version: 4.1.7 • Splunk Release Notes Version: 4.1.7 • Splunk Search Reference Version: 4.1.7 • Splunk User Manual Version: 4.1.7	
ALC_CMC.2 Use of a CM system	• Common Criteria Configuration Management (dated 11/30/10) • 4.1.7-93600-P4-Repository.txt	This document describes the authorization controls for the TOE.
ALC_CMS.2 Parts of the TOE CM coverage	• Common Criteria Configuration Management (dated 11/30/10) • 4.1.7-93600-P4-Repository.txt	These documents describe the CM scope of the TOE.
ALC_DEL.1 Delivery Procedures	Delivery_1.4_10252010.docx	This document describes product delivery for the TOE and a description of all procedures used to ensure objectives are not compromised in the delivery process.
ALC_FLR.2 Flaw reporting procedures	• Bug Triage Process (dated 12/13/10) • Flaw Remediation Document revision 0.4 • Jira (dated 12/13/10) • Maintenance Release Process (dated 12/13/10)	This document describes the processes taken for flaw remediation for the TOE.
ASE_CCL.1 Conformance Claims	Splunk 4.1.7 Security Target 2.0	This document describes the CC conformance claims made by the TOE.
ASE_ECD.1 Extended Components Definition	Splunk 4.1.7 Security Target 2.0	This document provides a definition for all extended components in the TOE.
ASE_INT.1 Security Target Introduction	Splunk 4.1.7 Security Target 2.0	This document describes the Introduction of the Security Target.
ASE_OBJ.2	Splunk 4.1.7 Security Target 2.0	This document describes all of the security objectives for the

Component	Document(s)	Rationale
Security Objectives		TOE.
ASE_REQ.2 Derived Security Requirements	Splunk 4.1.7 Security Target 2.0	This document describes all of the security requirements for the TOE.
ASE_SPD.1 Security Problem Definition	Splunk 4.1.7 Security Target 2.0	This document describes the security problem definition of the Security Target.
ASE_TSS.1 TOE Summary Specification	Splunk 4.1.7 Security Target 2.0	This document describes the TSS section of the Security Target.
ATE_COV.1 Evidence of Coverage	• splunk_test_report-2010-11-18-linuxsol.doc • splunk_test_report-2010-11-18-win.doc • FMT-FDP Test Tables-Results.xlsx • FMT-FDP_Test_Tables_CentOS_5.3_x64_FF3.6_Results.xlsx • Splunk diag files.zip	This document provides an analysis of coverage for the TOE.
ATE_FUN.1 Functional Testing	• splunk_test_report-2010-11-18-linuxsol.doc • splunk_test_report-2010-11-18-win.doc • FMT-FDP Test Tables-Results.xlsx • FMT-FDP_Test_Tables_CentOS_5.3_x64_FF3.6_Results.xlsx • Splunk diag files.zip	This document describes the functional tests for the TOE.
ATE_IND.2 Independent Testing - sample	Splunk 4.1.7 Evaluation Team Test Report version 2.0	This document describes the independent testing for the TOE.
AVA_VAN.2 Vulnerability Analysis	Splunk 4.1.7 Vulnerability Analysis version 1.0	This document describes the vulnerability analysis of the TOE.

Table 10-5: Assurance Requirements Evidence