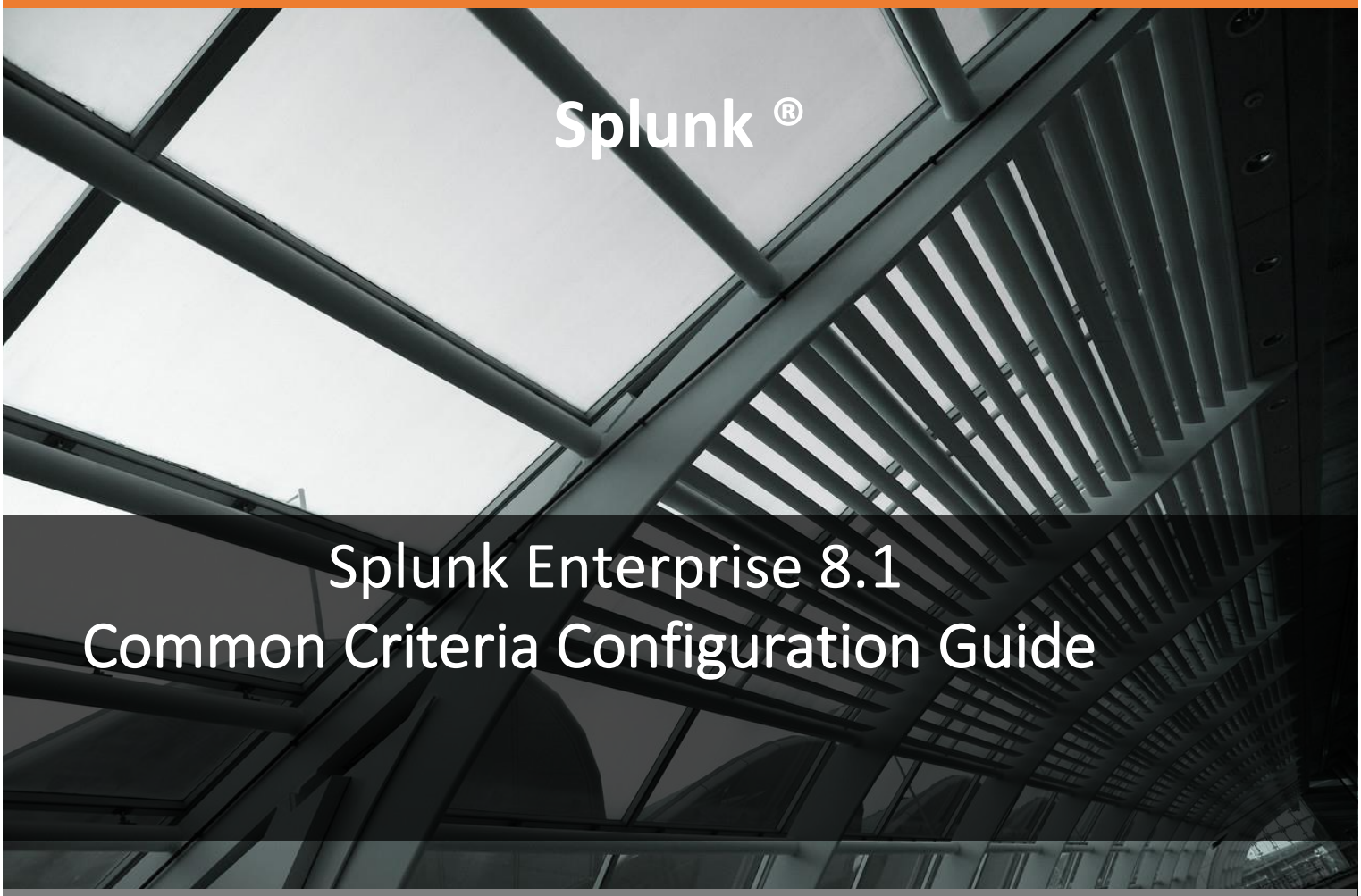


Splunk ®

Splunk Enterprise 8.1
Common Criteria Configuration Guide

Document Version: 0.9
Date: January 21, 2021

Prepared By:
Acumen Security
2400 Research Blvd Suite 395
Rockville, MD, 20850
www.acumensecurity.net

Table of Contents

Revision History:

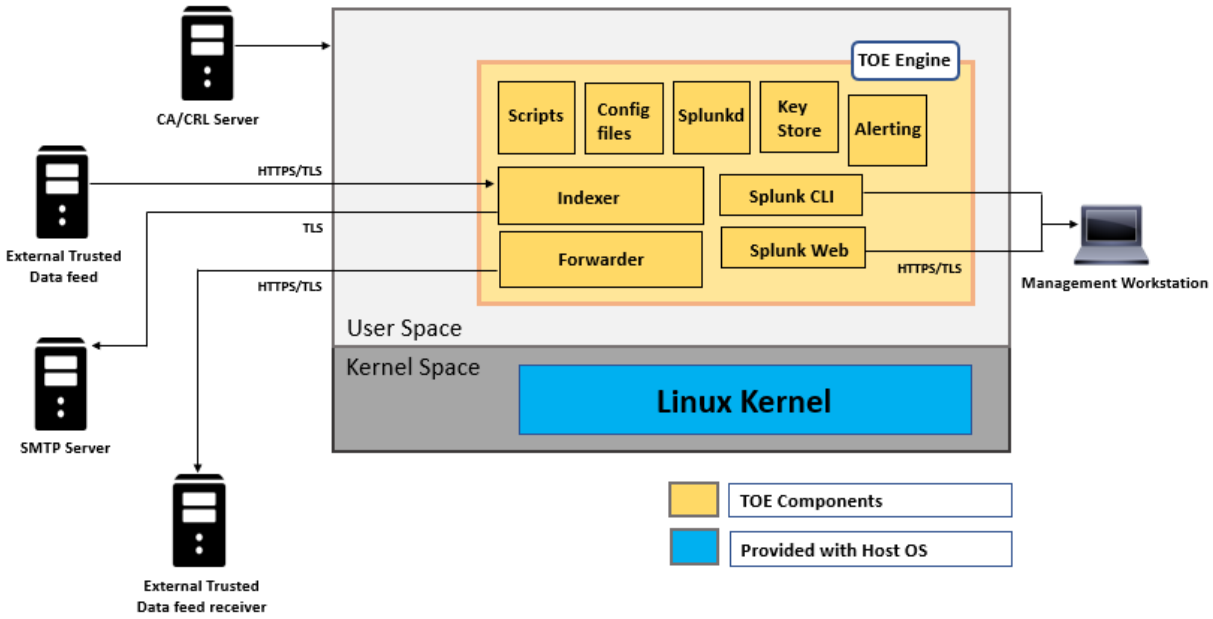
Version	Date	Changes
Version 0.1	05-22-2020	Initial Release
Version 0.2	07-28-2020	Updates based on Internal comments
Version 0.3	09-24-2020	Updates based on Internal review
Version 0.4	09-29-2020	Updates based on Internal review
Version 0.5	10-09-2020	Updates based on Internal review
Version 0.6	11-06-2020	Updates based on new build
Version 0.7	11-11-2020	Updates based on new build
Version 0.8	11-29-2020	Updates based on internal comments
Version 0.9	01-21-2021	Updates based on validator comments

1 Purpose of this document

This document is a guide to the Splunk Enterprise v8.1 implementation of the Common Criteria Application Protection Profile v1.3 (SWAPP v1.3). The information contained in this document is intended for Administrators who would be responsible for the configuration and management of the Splunk Enterprise 8.1 which runs on Red Hat Linux Enterprise (RHEL) v7.7 and v8.2 operating system.

This document will guide how to install, Configure, and operate the Application in a Common Criteria Compliant mode.

- Prerequisite for Installing Splunk Enterprise v8.1
- How to Install Splunk Enterprise v8.1 on RHEL 7.7 and RHEL 8.2
- The secure communication mechanisms employed Splunk Enterprise.
- How to Update the Splunk Enterprise.



1.1 TOE Overview

The Target of Evaluation (TOE) is the Splunk Enterprise v8.1 which runs on Red Hat Linux Enterprise (RHEL) v7.7 and v8.2 operating systems. Splunk collects data from various sources such as systems, devices, and interactions and presents the data for real time visibility and analysis. The TOE can be configured as a forwarder and an indexer. When the TOE is configured as the indexer, it will receive data from external sources such as web services, databases, and one or more instance of Splunk configured as a Forwarder. In Forwarder configuration, it will transmit all system generated data to the other instance of Splunk configured as an Indexer.

1.2 Target of Evaluation

The TOE is the Splunk Enterprise v8.1 which is executed on RHEL operating system. The Splunk Enterprise is a software application that enables users to search, analyze and visualize the data that is gathered from various components of an IT infrastructure or business industry. The Splunk Enterprise v8.1 conforms to the Common Criteria Application Protection Profile v1.3 dated 01 March 2019 [SWAPP]and Functional Package for Transport Layer Security (TLS), Version 1.1, dated 01 March 2019 [TLS-PKG].

2 TOE Description

2.1 Evaluated Configuration

The TOE is the Splunk Enterprise v8.1 which is executed on RHEL operating system. The Splunk Enterprise is a software application that enables users to search, analyze, and visualize the data that is gathered from various components of an IT infrastructure or business industry. The evaluated version of the TOE is v8.1.

In the evaluated configuration, the software was installed on the following hardware:

- Dell PowerEdge R430 Server with Intel Xeon E5-2630 v4 (Broadwell) processor

Note: The TOE is the application software only. The host platforms are not part of the evaluation.

The TOE supports secure connectivity with several other IT environment devices as described below.

Component	Required	Usage/Purpose Description
External Trusted Data Feed	Yes	External data source for transmitting non-TSF related data to the TOE indexer for populating Splunk's datastore. The external data source must use HTTPS/TLS to communicate with the TOE.
External Trusted Data Feed Receiver	Yes	External data source for receiving non-TSF related data from the TOE forwarder. The external data source must use HTTPS/TLS to communicate with the TOE.
Host Platform	Yes	A general-purpose computer on which the RHEL operating system and the TOE is installed.
Management Workstation	Yes	Used to remotely manage the TOE via HTTPS/TLS interface.
SMTP Server	Yes	External data source for receiving non-TSF related data from the TOE Indexer. The external data source must use HTTPS/TLS to communicate with the TOE.
CRL Server	Yes	Server which contains updated revocation list for the TOE.

2.2 Physical Boundaries

The TOE is a software application running on Dell PowerEdge R430 Server with Intel Xeon E5-2630 v4 (Broadwell) processor, and it includes 1TB disk and 32GB RAM. The TOE is Splunk Enterprise v8.1 which runs on Red Hat Linux Enterprise (RHEL) v7.7 and v8.2 64-bit operating system.

2.3 Logical Boundaries

The TOE provides the security functionality required by [SWAPP] and [TLS v1.1 package].

2.4 Other Assumptions

The following assumptions are drawn directly from the [SWAPP].

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

2.5 System Requirements

The systems that will host Splunk Enterprise v8.1 must meet the following requirements for Splunk to support Common Criteria compliance:

- Intel x86 64-bit chip architecture
- 10 CPU cores at 2Ghz or greater speed per core
- 12GB RAM
- Standard 64-bit RHEL7.7 and 8.2 with SElinux enabled
- Standard 1Gb Ethernet NIC for Network Connectivity

NOTE: The TOE does not require access to any sensitive repositories.

3 Splunk Installation and Configuration on RHEL7.7

3.1 Prerequisites for Splunk Enterprise v8.1

Below steps must be followed before installing Splunk Enterprise v8.1 in common criteria mode.

Ensure following resources are available prior to beginning installation of Splunk Enterprise v8.1

1. Verify the Red Hat Subscription Manager on the RHEL server, Valid Subscription should be present on the system, if subscription is not available, get one from the below link as it is one of the mandatory step for Splunk to be installed and configured in Common Criteria mode.
   https://access.redhat.com/products/red-hat-subscription-management

2. Acquire Splunk SELinux rpm file from the Splunk before starting the Installation. This .rpm file has the SELinux policies that enables the Splunk Enterprise to run in Common Criteria mode.

3. Set the SELinux in "Enforcing" mode for the Splunk to work in common criteria mode, Check the status of the SELinux with the below process:
   - cat /etc/selinux/config and set SELINUX= enforcing.
   - Execute getenforce and observe the result enforcing. Command setenforce 1 can be used to set the SElinux in enforcing mode.

4. Splunk uses Python of the underlying RHEL system for its working. Python version should match with the following version.

   ```
   $/usr/bin/python --version
   2.7.5
   ```

5. System dependencies for GNOME keyring and Python can be checked with the following.

   ```
   yum install gnome-keyring-devel
   yum install gnome-python2-gnomekeyring
   ```

6. Minimum two encrypted LUKS partition on the hard drive should be present for $SPLUNK_HOME and $SPLUNK_ETC. Size of the partition could be 150G for $SPLUNK_HOME and 25G for $SPLUNK_ETC. Follow the below links to setup LUKS.

   - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sec-encryption.

3.2 Install Splunk Enterprise v8.1

Once all the prerequisites are completed follow the below process to install Splunk Enterprise v8.1 on RHEL7.7

- Download Splunk Enterprise v8.1 from the below link.

  https://www.splunk.com/en_us/download/splunk-enterprise.html

  Note: A user account is needed to download the Splunk enterprise edition from the Splunk's official website and by default it comes with 60 days free evaluation license.

- Create a new user "splunk" on the server with the below command, "splunk" user is the user under which the Splunk application will be running.

  ```
  useradd splunk
  ```

- Install Splunk Enterprise v8.1 as 'root' user.

  ```
  rpm -i splunk-8.1.0 -<xxxxxxxxxxxx>-linux-2.6-x86_64.rpm
  ```

- After successful installation of Splunk change the location of Splunk's configuration files to /etc/opt/splunk.

  ```
  mv /opt/splunk/etc /etc/opt/splunk
  export SPLUNK_ETC=/etc/opt/splunk
  ```

- Install the Splunk SElinux rpm file acquired from the Splunk using the command "yum install splunk-SELinux -<version>.rpm"

- During Installation, a username and password prompt will be displayed. Use the same credentials for logging later to the Splunk UI. Password should be a mixture of uppercase, lowercase, numeric, and special characters for enhanced security.

  ```
  Please enter an administrator username: administrator
  Please enter a new password: *********
  Please confirm new password: *********
  ```

3.3 Automate Splunk with Scripts

Scripts can be created to automate functionalities in Splunk. Administrator can start or stop the Splunk application with the below scripts, These Scripts must be present under the home directory of Splunk user(/home/splunk).

- To start dbus create the "run_dbus.sh" script using below code.

  ```
  #!/bin/bash
  OUTPUT=$( runcon -t splunk_dbusd_t  -r system_r dbus-daemon --session --print-pid --print-address  --fork )
  echo $OUTPUT > /tmp/dbus-address
  export DBUS_SESSION_BUS_ADDRESS=$(awk '{ print $1}' /tmp/dbus-address)
  export DBUS_SESSION_BUS_PID=$(awk '{ print $2}' /tmp/dbus-address)
  ```

```
export PATH=/usr/bin:$PATH
echo $DBUS_SESSION_BUS_ADDRESS
echo $DBUS_SESSION_BUS_PID
```

- To stop dbus create the “stop_dbus.sh” script using below code.

```
#!/bin/bash
export DBUS_SESSION_BUS_PID=$(awk '{ print $2}' /tmp/dbus-address)
kill $DBUS_SESSION_BUS_PID
```

- To run the Splunk create the “run_splunk.sh” script using below code.

```
#!/bin/bash
export DBUS_SESSION_BUS_ADDRESS=$(awk '{ print $1}' /tmp/dbus-address)
export DBUS_SESSION_BUS_PID=$(awk '{ print $2}' /tmp/dbus-address)
export PATH=/usr/bin:$PATH
. /opt/splunk/bin/setSplunkEnv
runcon -u system_u -t splunk_t -r system_r splunk start
```

- To Stop the Splunk create the “stop_splunk.sh” script using below code.

```
#!/bin/bash
export DBUS_SESSION_BUS_ADDRESS=$(awk '{ print $1}' /tmp/dbus-address)
export DBUS_SESSION_BUS_PID=$(awk '{ print $2}' /tmp/dbus-address)
export PATH=/usr/bin:$PATH
. /opt/splunk/bin/setSplunkEnv
runcon -t splunk_t -r system_r splunk stop
```

- Perform the below steps to ensure the scripts has the correct SELinux context, below commands should be run as “root”

```
chown splunk:splunk /home/splunk/*
chcon -u system_u -r object_r  -t initrc_exec_t /home/splunk/run_*
chcon -u system_u -r object_r  -t initrc_exec_t /home/splunk/stop_*
chmod 755 /home/splunk/run_* /home/splunk/stop_*
chcon -u system_u -r object_r -t splunk_usr_t /home/splunk
```

- Run the below command as the user ‘splunk’ to set the correct user environment. splunk user is the user under which Splunk Enterprise works.

```
su - splunk
export SPLUNK_HOME=/opt/splunk
export SPLUNK_ETC=/etc/opt/splunk
```

3.4 Setting up Secure Communication in Splunk Enterprise v8.1

3.4.1 Considerations for CA-signed certificate

An administrator must configure and use certificates which can be generated via OpenSSL or any other tool from a different platform. All certificates must be FIPS-compliant, and CA-signed certificate chain should be issued by a trusted Certificate Authority. The CA-signed certificates must be in PEM format also should contain valid serial number and valid issuer domain name. The Basic Constraint CA flag must be set to True in case of root and intermediate certificates. For identity certificate, the flag must be set to False. The certificate must comply with the Extended Key Usage as "TLS WebServerAuthentication" and "TLS WebClientAuthentication", also KeyUSage must not be set to Critical. Ensure that the validity period for the certificate specifies a valid date range.

3.4.2 Digital Certificates

For securing communication in Splunk certificates are needed. Splunk does not generate certificates by its own in Common Criteria mode and will not have network communication. Web UI will not be functional, and the errors will be displayed under the Splunkd.log in the directory /opt/splunk/var/log/splunk.

OpenSSL or any other tool can be used to generate certificates. These certificates must be FIPS-compliant. In order to generate FIPS-complaint Certificates with Openssl OPENSSL_FIPS=1 keyword must be used for the Splunk to decrypt the certificate, absence of the OPENSSL_FIPS=1 keyword while generating certificates with Openssl can create errors and the errors will be logged under the Splunkd.log file. Certificates issued by CAs such as Verisign can also be used. The certificates must be in PEM format.

Certificates are needed for the communication between Splunk Indexer-Splunk Forwarder, Splunk Web UI-Browser and Splunk Indexer-SMTP server.

3.4.3 Configuration Files in Splunk

Splunk uses below configuration files to store the configuration needed for the communication between different entities. Administrator must create or modify these files under "/etc/opt/splunk/system/local" directory. In case of the absence of the below files new must be created.

- **server.conf**

```
[general]
requireBootPassphrase = true
allowRemoteLogin = never

[sslConfig]
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = <comma separated curves>
sendStrictTransportSecurityHeader = true
serverCert = <absolute_path_to_server_certificate>
sslAltNameToCheck = <comma_separated_list_of_SANs>
sslCommonNameList = <comma_separated_list_of_CNs>
sslRootCAPath = <path_to_the_CA>
sslVerifyServerCert = true
sslVersions = tls1.2
```

sslVersionsForClient = tls1.2

[kvstore]
serverCert = <absolute_path_to_kvstore_certificate>

[applicationsManagement]
allowInternetAccess = false

- **web.conf**

[settings]
cipherSuite= ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = <comma separated curves>
enableSplunkWebSSL = 1
privKeyPath = <absolute_path_to_encrypted_private_key>
serverCert = <absolute_path_to_public_certificate>
sslVersions = tls1.2
requireClientCert = true
sslAltNameToCheck = <identifier of client cert SAN>
sslCommonNameToCheck = <identifier of client cert CN>

- **authentication.conf**

[secrets]
disabled = false

- **alert_actions.conf**

[email]
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = <comma separated curves>
pdf.html_image_rendering = false
sslAltNameToCheck = <comma_separated_list_of_SANs>
sslCommonNameToCheck = <comma_separated_list_of_CNs>
sslVerifyServerCert = true
sslVersions = tls1.2
use_ssl = 1

Note: configure when using Splunk as an indexer for evaluated configuration.

- **inputs.conf**

[SSL]
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = <comma separated curves>

requireClientCert = true
serverCert = <absolute_path_to_server_cert>
sslAltNameToCheck = <comma_separated_list_of_SANs>
sslCommonNameToCheck = <comma_separated_list_of_CNs>
sslVersions = tls1.2

Note: configure when using Splunk as an Indexer, Indexer receives the data from the Forwarders.

- **outputs.conf**

[tcpout]
defaultGroup = group1
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = <comma separated curves>
clientCert = <path_to_client_certificate>
sslAltNameToCheck = <comma_separated_list_of_SANs>
sslCommonNameToCheck = <comma_separated_list_of_CNs>
sslVerifyServerCert = true
sslVersions = tls1.2
useClientSSLCompression = true

Note: configured when using Splunk as a Forwarder, Forwarders sends the data to the indexer.

3.4.4 Enable Secret Store in Splunk

Splunk uses GNOME keyring to store key data. Private keys or other secret data must be stored under the LUKS encrypted partition.

Splunk requires dbus daemon to be Up and running and the dbus daemon is needed for communication with GNOME keyring. Start the dbus process using below script. For further details on script created for dbus process refer section 3.3.

#/home/splunk/run_dbus.sh

Verify that dbus process is running in splunk_dbusd_t SELinux context. Please use the below command.

#ps auxZ | grep dbus

An output of the about command will appear as below.

#unconfined_u:system_r:splunk_dbusd_t:s0-s0:c0.c1023 splunk 27374 0.0 0.0 68400 1652 ? Ssl Apr20 0:00 dbus-daemon --session --print-pid --print-address –fork

- To Initialize secret storage password, use the below command.

#runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin splunk secret-storage -- unlock

- To view the list of keys available for secret storage issue the below command.

```
#runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin splunk secret-storage -- spec
```

3.4.5 Storing Secrets in GNOME keyring

Use the below command to add secrets to the GNOME keyring.

```
#runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin splunk secret-storage --write --
no-prompt <conf-file> <stanza-name> <attribute-name> <passphrase>
```

Conf-file (e.g. inputs.conf) stanza-name: name of stanza (e.g. sslConfig) attribute-name: name of attribute (e.g. sslKeysfilePassword) passphrase: passphrase to be used

3.4.6 Enable CC mode in Splunk

Add the below lines in the splunk-launch.conf file under the directory /etc/opt/splunk to enable common criteria mode in Splunk.

```
SPLUNK_COMMON_CRITERIA=1
SPLUNK_FIPS=1
```

Append the /home/splunk/.bashrc file with the below content so that environment is setup properly.

```
export SPLUNK_ETC=/etc/opt/splunk
export DBUS_SESSION_BUS_ADDRESS=$(awk '{ print $1}' /tmp/dbus-address)
export DBUS_SESSION_BUS_PID=$(awk '{ print $2}' /tmp/dbus-address)
export PATH=/usr/bin:$PATH
. /opt/splunk/bin/setSplunkEnv
```

3.4.7 Configure Splunk to Communicate with SMTP server over SSL/TLS

Splunk can be configured to communicate with SMTP server over SSL/TLS for sending email alerts. Alerts are the actions which gets triggered when a specific criterion is met as defined by the user. Splunk can be configured to send these alerts via email to the email server. Use the below configuration to send email from the Splunk to the SMTP server.

Configure alert_actions.conf file in the directory /etc/opt/splunk/system/local on the Splunk (mostly Indexer) to send the email alerts.
alert_actions.conf

```
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384
pdf.html_image_rendering = 0
sslCommonNameToCheck = <common name to check>
sslAltNameToCheck = <alternate name to check>
sslVerifyServerCert = true
sslVersions = tls1.2
use_ssl = 1
mailserver = <SMTP server ip>:<port>
auth_username = email username
```

Below search in the GUI can be used for sending email alert to the email server.

```
index="_internal" | head 5 | sendemail to="username@splunk.com" subject="Here is an email notification" message="This is an example message" sendresults=true inline=true format=raw use_ssl=true
```

3.4.8 Configure Mutual Authentication between Indexer and Forwarder over HTTPS/TLS

Splunk Indexer receives the logs from the forwarder and parse these logs to show them on the Splunk dashboard. Indexer and the forwarder authenticate each other using digital certificates. Following constraint must be configured in the inputs.conf file to enable mutual authentication:

- requireClientCert is set to true

Below configuration is required on both the devices for the communication to be successful.

- On Indexer (Inputs.conf)

```
[SSL]
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = prime256v1,secp384r1,secp521r1
requireClientCert = true
serverCert = /etc/opt/splunk/auth/Inputs.pem
sslAltNameToCheck = <common name to check>
sslCommonNameToCheck = <alternate name to check>
sslPassword = <Private key password or can be added via secret storage>
sslVersions = tls1.2

[splunktcp-ssl://9998]
disabled = 0
```

Note: secp256r1 is equivalent to prime256v1.

- On Forwarder (outputs.conf)

```
[tcpout]
defaultGroup = group1

[tcpout:group1]
server = <ip address of indexer>:<port>
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
clientCert = /etc/opt/splunk/auth/Output.pem
sslAltNameToCheck = <alternate name to check>
sslCommonNameToCheck = <common name to check>
sslVerifyServerCert = true
sslVersions = tls1.2
sslPassword = <Private key password or can be added via secret storage>
```

useClientSSLCompression = true

3.4.9 Configure Splunk Web UI to Communicate using Mutual Authentication

Splunk web configuration file is required to access the Splunk Web UI in common criteria mode. Splunk Administrator must create/modify "web.conf" file under directory /etc/opt/splunk/system/local for the WEB UI to work. Following constraint must be configured to enable mutual authentication:

- requireClientCert is set to true

On Indexer and Forwarder (web.conf)

[settings]
cipherSuite= ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
enableSplunkWebSSL = 1
privKeyPath = /etc/opt/splunk/auth/splunkweb.key
serverCert = /etc/opt/splunk/auth/splunkweb.pem
sslVersions = tls1.2
requireClientCert = true
sslAltNameToCheck = <alternate name to check>
sslCommonNameToCheck = <common name to check>

3.4.10 Configure Server Configuration file on Indexer and Forwarder

Server configuration file contains a wide variety of settings that supports the overall state of a Splunk Enterprise instance. For example, server.conf file includes settings for enabling SSL, configuring nodes of an indexer cluster or a search head cluster, configuring KV store, and setting up a license master.

- On indexer and forwarder (server.conf)

[general]
requireBootPassphrase = true
allowRemoteLogin = never

[sslConfig]
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = prime256v1,secp384r1,secp521r1
sendStrictTransportSecurityHeader = true
serverCert = /etc/opt/splunk/auth/server.pem
sslAltNameToCheck = <alternate name to check>
sslCommonNameToCheck = <common name to check>
sslRootCAPath = <path to CA>
sslVerifyServerCert = true
sslVersions = tls1.2
sslVersionsForClient = tls1.2
sslPassword = <Private key password or can be added via secret storage>

```
[kvstore]
serverCert = /etc/opt/splunk/auth/splunkkv.pem
sslPassword = <Private key password or can be added via secret storage>

[applicationsManagement]
allowInternetAccess = false
```

Note: secp256r1 is equivalent to prime256v1.

3.4.11 Configure Certification Revocation list (CRL) in Splunk Enterprise v8.1

Splunk expects CRLs for revocation checking under directory "/etc/opt/splunk/auth/crl" in PEM format. CRL can be directly placed under this directory or can be automated by creating a below script.

```
#cat crl.txt
http://crl server ip or domain-name/filename.der
```

The bash script shown below can be configured which will look in to the crl.txt file and download the CRL files under the directory /etc/opt/splunk/auth/crl and convert them into PEM format if needed.

```
#!/bin/bash

# NOTE: Only applicable for Splunk version 8.1.x, while running in Common Criteria mode.
# This script is provided as an example for downloading the CRL files in a location
# Splunk expects it to be. Any other mechanism which updates CRL files should work.
# The user can run the script one time OR setup a cron job to run it periodically (say every 30 min).
# The script cleans out ALL existing CRL files (*.crl, *.pem) and then downloads the new versions.

# Example invocation: /home/splunk/update_crl.sh /home/splunk/crl.txt /etc/opt/splunk/auth/crl

if [ "$#" -ne 2 ]; then
   echo "Usage: $0 <crllist_file_absolute_path> <crl_download_location_absolute_path>"
   exit 1
fi

PWD=`pwd`

filename=$1
crl_dir=$2

if [ ! -f "$filename" ] || [ ! -d "$crl_dir" ] || [[ "$filename" != /* ]] || [[ $crl_dir != /* ]]; then
   echo "Both the crllist_file and crl_download_location must exist and be specified as
absolute paths."
   exit 2
fi

# go to $crl_dir
```

```
cd crl_dir

# remove older CRL files if present
rm -rf *.crl *.pem

while read -r line || [[ -n "$line" ]]; do
  url=$line
  wget $url
  if [ "$?" -ne 0 ]; then
        echo "Failed to download CRL file: $url"
  fi
done < "$filename"

# For each file except README in this dir, check if the file is in DER format.
# If yes, then convert to PEM and remove the corresponding CRL file.
for f in ./*
do
  if [ $f != "./README" ];then
# use openssl from the OS itself
        openssl crl -in $f -text -noout &> /dev/null
        if [ "$?" -ne 0 ]; then #DER format, must convert to PEM
              openssl crl -inform der -in $f -out $f.pem
              if [ "$?" -ne 0 ]; then
                    echo "Failed to convert DER format CRL file ($f) into PEM
format. Splunk will not use this CRL file"
              fi
              rm $f
        fi
  fi
done

#revert to old pwd
 cd $PWD
```

The above script can be stored under the directory /home/splunk/update_crl.sh with the appropriate SELinux context and file-permissions.

```
chown splunk:splunk update_crl.sh crl.txt
chcon -u system_u -r object_r -t initrc_exec_t update_crl.sh
chmod 755 update_crl.sh
```

3.4.12 Removing Binaries

Following binaries needs to be removed post installation of the TOE for the configuration to be CC complaint as these binaries lacks stack-based buffer overflow support (canary detector tool was used to check the support).

- rm /opt/splunk/bin/mongodump
- rm /opt/splunk/bin/mongorestore

- rm /opt/splunk/bin/srm

NOTE: Please note that removal of the above binaries do not affect functionality of Splunk.

3.5 Start Splunk and Validate Configuration

- Start the Splunk with "run_splunk.sh" script created in Section 3.3

  /home/splunk/run_splunk.sh

- Check that the Splunk is running with the splunk SELinux context:

  ps auxZ | grep splunk

- To verify that Splunk is in Common Criteria mode, check the splunkd.log file in the directory /opt/splunk/var/log/splunk and look for the following message which verifies Splunk is running in Common Criteria Mode.

  ServerConfig - Splunk is starting in Common Criteria Mode.

- Below Splunk UI will be available and login with the credentials that were created in the earlier part of this document.

4 Splunk Installation and Configuration on RHEL8.2

4.1 Prerequisites for Splunk Enterprise v8.1

Below steps must be followed before installing Splunk Enterprise v8.1 in common criteria mode.

Ensure following resources are available prior to beginning installation of Splunk Enterprise v8.1

1. Verify the Red Hat Subscription Manager on the RHEL server, Valid Subscription should be present on the system, if subscription is not available, get one from the below link as it is one of the mandatory step for Splunk to be installed and configured in Common Criteria mode.
   https://access.redhat.com/products/red-hat-subscription-management

2. Acquire Splunk SELinux rpm file from the Splunk before starting the Installation. This .rpm file has the SELinux policies that enables the Splunk Enterprise to run in Common Criteria mode.

3. Set the SELinux in "Enforcing" mode for the Splunk to work in common criteria mode, Check the status of the SELinux with the below process:
   - cat /etc/selinux/config and set SELINUX= enforcing.
   - Execute getenforce and observe the result enforcing. Command setenforce 1 can be used to set the SElinux in enforcing mode.

4. Splunk uses Python provided by RHEL for its working. Force the python version to python3 with below command

   $alternatives --set python /usr/bin/python3

5. Acquire Gnome keyring python script from the Splunk before starting the Installation.

6. Minimum two LUKS encrypted partition on the hard drive should be present for $SPLUNK_HOME and $SPLUNK_ETC. Size of the partition could be 150G for $SPLUNK_HOME and 25G for $SPLUNK_ETC. Follow the below links to setup LUKS.

   - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sec-encryption.

4.2 Install Splunk Enterprise v8.1

Once all the prerequisites are completed follow the below process to install Splunk Enterprise version v8.1 on RHEL8.2

- Download Splunk Enterprise v8.1 from the below link.

  https://www.splunk.com/en_us/download/splunk-enterprise.html

Note: A user account is needed to download the Splunk enterprise edition from the Splunk's official website and by default it comes with 60 days free evaluation license.

- Create a new user "splunk" on the server with the below command, "splunk" user is the user under which the Splunk application will be running.

```
useradd splunk
```

- Install Splunk Enterprise v8.1 as 'root' user.

```
rpm -i splunk-8.1.0 -<xxxxxxxxxxxx>-linux-2.6-x86_64.rpm
```

- After successful installation of Splunk change the location of Splunk's configuration files to /etc/opt/splunk.

```
mv /opt/splunk/etc /etc/opt/splunk
export SPLUNK_ETC=/etc/opt/splunk
```

- Install the Splunk SElinux rpm file acquired from the Splunk using the command "yum install splunk-SELinux -<version>.rpm"

- Copy the acquired Gnome keyring python script in the Splunk directory /etc/opt/splunk/system/bin.

4.3 Setting up Secure Communication in Splunk Enterprise v8.1

4.3.1 Considerations for CA-signed certificate

An administrator must configure and use certificates which can be generated via OpenSSL or any other tool from a different platform. All certificates must be FIPS-compliant, and CA-signed certificate chain should be issued by a trusted Certificate Authority. The CA-signed certificates must be in PEM format also should contain valid serial number and valid issuer domain name. The Basic Constraint CA flag must be set to True in case of root and intermediate certificates. For identity certificate, the flag must be set to False. The certificate must comply with the Extended Key Usage as "TLS WebServerAuthentication" and "TLS WebClientAuthentication", also KeyUSage must not be set to Critical. Ensure that the validity period for the certificate specifies a valid date range.

4.3.2 Digital Certificates

For securing communication in Splunk certificates are needed. Splunk does not generate certificates by its own in Common Criteria mode and will not have network communication. Web UI will not be functional, and the errors will be displayed under the Splunkd.log in the directory /opt/splunk/var/log/splunk.

OpenSSL or any other tool can be used to generate certificates. These certificates must be FIPS-compliant. In order to generate FIPS-complaint Certificates with Open SSL OPENSSL_FIPS=1 keyword must be used for the Splunk to decrypt the certificate, absence of the OPENSSL_FIPS=1 keyword while generating certificates with Open SSL can create errors and the errors will be logged under the Splunkd.log file. Certificates issued by CAs such as Verisign can also be used. The certificates must be in PEM format.

NOTE: Splunk will not have network communication with default certificates. Openssl version shipped with RHEL 8.2 cannot generate FIPS-Compliant certificates, one can use openssl version which is shipped with Splunk to generate FIPS-compliant certificates.

Below procedure should be followed to generate FIPS-compliant private key with Splunk's openssl version. These commands should be run as a 'splunk' user.

```
export OPENSSL_FIPS=1
export SPLUNK_HOME=/opt/splunk
export SPLUNK_ETC=/etc/opt/splunk
openssl ecparam -name prime256v1 -genkey -noout -out splunk.key
OPENSSL_FIPS=1 runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk cmd openssl
OpenSSL> ec -aes256 -in splunk.key -out splunk.enc.key
```

Certificates are needed for the communication between Splunk Indexer-Splunk Forwarder, Splunk Web UI-Browser and Splunk Indexer-SMTP server.

Note: secp256r1 is equivalent to prime256v1.

4.3.3 Configuration Files in Splunk

Splunk uses below configuration files to store the configuration needed for the communication between different entities. Administrator must create or modify these files under "/etc/opt/splunk/system/local" directory. In case of the absence of the below files new must be created.

- **server.conf**

```
[general]
requireBootPassphrase = true
allowRemoteLogin = never

[sslConfig]
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = <comma separated curves>
sendStrictTransportSecurityHeader = true
serverCert = <absolute_path_to_server_certificate>
sslAltNameToCheck = <comma_separated_list_of_SANs>
sslCommonNameList = <comma_separated_list_of_CNs>
sslRootCAPath = <path_to_the_CA>
sslVerifyServerCert = true
sslVersions = tls1.2
sslVersionsForClient = tls1.2

[kvstore]
serverCert = <absolute_path_to_kvstore_certificate>

[applicationsManagement]
allowInternetAccess = false
```

- **web.conf**

```
[settings]
```

cipherSuite= ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = <comma separated curves>
enableSplunkWebSSL = 1
privKeyPath = <absolute_path_to_encrypted_private_key>
serverCert = <absolute_path_to_public_certificate>
sslVersions = tls1.2
requireClientCert = true
sslAltNameToCheck = <identifier of client cert SAN>
sslCommonNameToCheck = <identifier of client cert CN>

- **authentication.conf**

  [secrets]
  disabled = false

- **alert_actions.conf**

  [email]
  cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
  ecdhCurves = <comma separated curves>
  pdf.html_image_rendering = false
  sslAltNameToCheck = <comma_separated_list_of_SANs>
  sslCommonNameToCheck = <comma_separated_list_of_CNs>
  sslVerifyServerCert = true
  sslVersions = tls1.2
  use_ssl = 1

  Note: configure when using Splunk as an indexer for sending email alerts to SMTP server.

- **inputs.conf**

  [SSL]
  cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
  ecdhCurves = <comma separated curves>
  requireClientCert = true
  serverCert = <absolute_path_to_server_cert>
  sslAltNameToCheck = <comma_separated_list_of_SANs>
  sslCommonNameToCheck = <comma_separated_list_of_CNs>
  sslVersions = tls1.2

  Note: configure when using Splunk as an Indexer, Indexer receives the data from the Forwarders.

- **outputs.conf**

  [tcpout]

defaultGroup = group1
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = <comma separated curves>
clientCert = <path_to_client_certificate>
sslAltNameToCheck = <comma_separated_list_of_SANs>
sslCommonNameToCheck = <comma_separated_list_of_CNs>
sslVerifyServerCert = true
sslVersions = tls1.2
useClientSSLCompression = true

Note: configured when using Splunk as a Forwarder, Forwarders sends the data to the indexer.

4.3.4 Enable Secret Store in Splunk

Splunk uses GNOME keyring to store key data. Private keys or other secret data must be stored under the LUKS encrypted partition.

Splunk requires dbus daemon to be Up and running and the dbus daemon is needed for communication with GNOME keyring. Please start the dbus process using below command.

```
#sudo -Hu splunk runcon -u system_u -t splunk_t -r system_r dbus-run-session -- bash
```

Verify that dbus process is running in splunk_dbusd_t SELinux context. Please use the below command.

```
#ps auxZ | grep dbus
```

An output of the above command will appear as below.

```
# system_u:system_r:splunk_dbusd_t:s0-s0:c0.c1023 splunk 170722 0.0  0.0 82592 5004
? Sl Sep11   0:00 dbus-daemon --nofork --print-address 4 --session
```

- To Initialize secret storage password, Follow the below process.

```
#gnome-keyring-daemon –unlock
#Enter secret store password
#ctrl+d
```

An output of the above command will appear as below.

```
GNOME_KEYRING_CONTROL=/home/splunk/.cache/keyring-3IAHR0
SSH_AUTH_SOCK=/home/splunk/.cache/keyring-3IAHR0/ssh
```

- To set the Environmental variables, use the below commands.

```
export SPLUNK_ETC=/etc/opt/splunk
export SPLUNK_HOME=/opt/splunk
```

- To view the list of keys available for secret storage issue the below command.

  #runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin splunk secret-storage -- spec

4.3.5 Storing Secrets in GNOME keyring

Use the below command to add secrets to the GNOME keyring.

```
#runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk secret-storage --write --
no-prompt <conf-file> <stanza-name> <attribute-name> <passphrase>
```

Conf-file (e.g. inputs.conf) stanza-name: name of stanza (e.g. sslConfig) attribute-name: name of attribute (e.g. sslKeysfilePassword) passphrase: passphrase to be used

4.3.6 Enable CC mode in Splunk

Add the below lines in the splunk-launch.conf file under the directory /etc/opt/splunk to enable common criteria mode in Splunk.

```
SPLUNK_COMMON_CRITERIA=1
SPLUNK_FIPS=1
```

4.3.7 Configure Splunk to Communicate with SMTP server over SSL/TLS

Splunk can be configured to communicate with SMTP server over SSL/TLS for sending email alerts. Alerts are the actions which gets triggered when a specific criterion is met as defined by the user. Splunk can be configured to send these alerts via email to the email server. Use the below configuration to send email from the Splunk to the SMTP server.

Configure alert_actions.conf file in the directory /etc/opt/splunk/system/local on the Splunk (mostly Indexer) to send the email alerts.
alert_actions.conf

```
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384
pdf.html_image_rendering = 0
sslCommonNameToCheck = <common name to check>
sslAltNameToCheck = <alternate name to check>
sslVerifyServerCert = true
sslVersions = tls1.2
use_ssl = 1
mailserver = <SMTP server ip>:<port>
auth_username = email username
```

Below search in the GUI can be used for sending email alert to the email server.

```
index="_internal" | head 5 | sendemail to="username@splunk.com" subject="Here is an
email notification" message="This is an example message" sendresults=true inline=true
format=raw use_ssl=true
```

4.3.8 Configure Mutual Authentication between Indexer and Forwarder over HTTPS/TLS

Splunk Indexer receives the logs from the forwarder and parse these logs to show them on the Splunk dashboard. Indexer and the forwarder authenticate each other using digital certificates. Following constraint must be configured in the inputs.conf file to enable mutual authentication:

- requireClientCert is set to true.

Below configuration is required on both the devices for the communication to be successful.

- On Indexer (Inputs.conf)

```
[SSL]
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = prime256v1,secp384r1,secp521r1
requireClientCert = true
serverCert = /etc/opt/splunk/auth/Inputs.pem
sslAltNameToCheck = <common name to check>
sslCommonNameToCheck = <alternate name to check>
sslPassword = <Private key password or can be added via secret storage>
sslVersions = tls1.2

[splunktcp-ssl://9998]
disabled = 0
```

Note: secp256r1 is equivalent to prime256v1.

- On Forwarder (outputs.conf)

```
[tcpout]
defaultGroup = group1

[tcpout:group1]
server = <ip address of indexer>:<port>
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
clientCert = /etc/opt/splunk/auth/Output.pem
sslAltNameToCheck = <alternate name to check>
sslCommonNameToCheck = <common name to check>
sslVerifyServerCert = true
sslVersions = tls1.2
sslPassword = <Private key password or can be added via secret storage>
useClientSSLCompression = true
```

4.3.9 Configure Splunk Web UI to Communicate using Mutual Authentication

Splunk web configuration file is required to access the Splunk Web UI in common criteria mode. Splunk Administrator must create/modify “web.conf” file under directory /etc/opt/splunk/system/local. Following constraint must be configured to enable mutual authentication:

- requireClientCert is set to true.

- On Indexer and Forwarder (web.conf)

```
[settings]
cipherSuite= ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
enableSplunkWebSSL = 1
privKeyPath = /etc/opt/splunk/auth/splunkweb.key
serverCert = /etc/opt/splunk/auth/splunkweb.pem
sslVersions = tls1.2
requireClientCert = true
sslAltNameToCheck = <alternate name to check>
sslCommonNameToCheck = <common name to check>
```

4.3.10 Configure Server Configuration file on Indexer and Forwarder

Server configuration file contains a wide variety of settings that supports the overall state of a Splunk Enterprise instance. For example, server.conf file includes settings for enabling SSL, configuring nodes of an indexer cluster or a search head cluster, configuring KV store, and setting up a license master.

- On indexer and forwarder (server.conf)

```
[general]
requireBootPassphrase = true
allowRemoteLogin = never

[sslConfig]
cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
ecdhCurves = prime256v1,secp384r1,secp521r1
sendStrictTransportSecurityHeader = true
serverCert = /etc/opt/splunk/auth/server.pem
sslAltNameToCheck = <alternate name to check>
sslCommonNameToCheck = <common name to check>
sslRootCAPath = <path to CA>
sslVerifyServerCert = true
sslVersions = tls1.2
sslVersionsForClient = tls1.2
sslPassword = <Private key password or can be added via secret storage>

[kvstore]
serverCert = /etc/opt/splunk/auth/splunkkv.pem
sslPassword = <Private key password or can be added via secret storage>

[applicationsManagement]
allowInternetAccess = false
```

Note: secp256r1 is equivalent to prime256v1.

4.3.11 Configure Certification Revocation List (CRL) in Splunk v8.1

Splunk expects CRLs for revocation checking under directory "/etc/opt/splunk/auth/crl" in PEM format. CRL can be directly placed under this directory or can be automated by creating a below script.

```
#cat crl.txt
http://crl server ip or domain-name/filename.der
```

The bash script shown below can be configured which will look in to the crl.txt file and download the CRL files under the directory /etc/opt/splunk/auth/crl and convert them into PEM format if needed.

```
#!/bin/bash

# NOTE: Only applicable for Splunk version 8.1.x, while running in Common Criteria mode.
# This script is provided as an example for downloading the CRL files in a location
# Splunk expects it to be. Any other mechanism which updates CRL files should work.
# The user can run the script one time OR setup a cron job to run it periodically (say every 30 min).
# The script cleans out ALL existing CRL files (*.crl, *.pem) and then downloads the new versions.

# Example invocation: /home/splunk/update_crl.sh /home/splunk/crl.txt /etc/opt/splunk/auth/crl

if [ "$#" -ne 2 ]; then
  echo "Usage: $0 <crllist_file_absolute_path> <crl_download_location_absolute_path>"
  exit 1
fi

PWD=`pwd`

filename=$1
crl_dir=$2

if [ ! -f "$filename" ] || [ ! -d "$crl_dir" ] || [[ "$filename" != /* ]] || [[ $crl_dir != /* ]]; then
  echo "Both the crllist_file and crl_download_location must exist and be specified as
absolute paths."
  exit 2
fi

# go to $crl_dir

cd crl_dir

# remove older CRL files if present
rm -rf *.crl *.pem

while read -r line || [[ -n "$line" ]]; do
  url=$line
```

```
  wget $url
  if [ "$?" -ne 0 ]; then
        echo "Failed to download CRL file: $url"
  fi
done < "$filename"

# For each file except README in this dir, check if the file is in DER format.
# If yes, then convert to PEM and remove the corresponding CRL file.
for f in ./*
do
  if [ $f != "./README" ];then
# use openssl from the OS itself
        openssl crl -in $f -text -noout &> /dev/null
        if [ "$?" -ne 0 ]; then #DER format, must convert to PEM
              openssl crl -inform der -in $f -out $f.pem
              if [ "$?" -ne 0 ]; then
                    echo "Failed to convert DER format CRL file ($f) into PEM
format. Splunk will not use this CRL file"
              fi
              rm $f
        fi
  fi
done

#revert to old pwd
 cd $PWD
```

The above script can be stored under the directory /home/splunk/update_crl.sh with the appropriate SELinux context and file-permissions.

```
chown splunk:splunk update_crl.sh crl.txt
chcon -u system_u -r object_r -t initrc_exec_t update_crl.sh
chmod 755 update_crl.sh
```

4.3.12 Removing Binaries

Following binaries needs to be removed post installation of the TOE for the configuration to be CC complaint as these binaries lacks stack-based buffer overflow support (canary detector tool was used to check the support).

- rm /opt/splunk/bin/mongodump
- rm /opt/splunk/bin/mongorestore
- rm /opt/splunk/bin/srm

NOTE: Please note that removal of the above binaries do not affect functionality of Splunk.

4.4 Start Splunk and Validate Configuration

- Start the Splunk with the below command.

  runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk start

  During the first boot Splunk will prompt for the license agreement as below:

  SPLUNK GENERAL TERMS (v1.2020)

  Do you agree with this license? [y/n]:

  Type y and press enter.

  Next set the username and password that will be used for login later into Splunk UI.

  ```
  Please enter an administrator username: administrator
  Password must contain at least:
    * 8 total printable ASCII character(s).
  Please enter a new password: *********
  Please confirm new password: ********
  ```

- To verify that Splunk is in Common Criteria mode, check the splunkd.log in the directory /opt/splunk/var/log/splunk and look for the following message which verifies Splunk is running in Common Criteria Mode.

  ServerConfig - Splunk is starting in Common Criteria Mode.

- Below Splunk UI will be available and login with the credentials that were set in the above step.

5 Cryptographic Support

5.1.1 TLS Configuration

TLS configuration is required for the secure communication between the TOE and different entities. Administrator can manually configure the parameters like cipher suites, ssl version, curves, path to server certificate, path to client certificate, path to CA certificate and SAN/CN reference identifiers.

In case of Mutual Authentication make sure that inputs.conf and web.conf is configured with the following constraint:

- requireClientCert is set to “true”

The following ciphers are supported by the TOE:

- TLS_ECDHE_ECDSA_AES256-GCM-SHA384
- TLS_ECDHE_ECDSA_AES256-CBC-SHA384
- TLS_ECDHE_ECDSA_AES128-GCM-SHA256

The TOE only supports TLS 1.2, all other protocols such as SSL 2.0, SSL 3.0, TLS 1.0, and TLS 1.1 is rejected by the TOE. The TOE supports the following Elliptic Curves:

- secp256r1,
- secp384r1,
- secp521r1

Note: secp256r1 is equivalent to prime256v1.

Below .conf files can be modified to configure TLS parameters. Following constrains must be used to configure these parameters:

cipherSuite is set to “ciphers supported”
ecdhCurves is set to “supported curves”
serverCert is set to “full path to server cert”
clientCert is set to “full path to client cert”
sslVersions is set to “supported tls version”
sslRootCAPath is set to “full path to CA cert”
sslAltNameToCheck is set to “list of SAN entries in the certificate”
sslCommonNameList is set to “list of common names in the certificate”

- **server.conf**

  [sslConfig]
  cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
  ecdhCurves = <comma separated curves>
  serverCert = <absolute_path_to_server_certificate>
  sslAltNameToCheck = <comma_separated_list_of_SANs>
  sslCommonNameList = <comma_separated_list_of_CNs>
  sslRootCAPath = <path_to_the_CA>

sslVerifyServerCert = true
sslVersions = tls1.2

- **alert_actions.conf**

  [email]
  cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
  ecdhCurves = <comma separated curves>
  sslAltNameToCheck = <comma_separated_list_of_SANs>
  sslCommonNameToCheck = <comma_separated_list_of_CNs>
  sslVersions = tls1.2

  Note: configure when using Splunk as an indexer for sending email alerts to SMTP server.

- **inputs.conf**

  [SSL]
  cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
  ecdhCurves = <comma separated curves>
  requireClientCert = true
  serverCert = <absolute_path_to_server_cert>
  sslAltNameToCheck = <comma_separated_list_of_SANs>
  sslCommonNameToCheck = <comma_separated_list_of_CNs>
  sslVersions = tls1.2

  Note: configure when using Splunk as an Indexer, Indexer receives the data from the Forwarders.

- **outputs.conf**

  [tcpout]
  defaultGroup = group1
  cipherSuite = ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384
  ecdhCurves = <comma separated curves>
  clientCert = <path_to_client_certificate>
  sslAltNameToCheck = <comma_separated_list_of_SANs>
  sslCommonNameToCheck = <comma_separated_list_of_CNs>
  sslVersions = tls1.2

  Note: configured when using Splunk as a Forwarder, Forwarders sends the data to the indexer.

- **web.conf**

  [settings]
  cipherSuite= ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384

ecdhCurves = <comma separated curves>
serverCert = <absolute_path_to_public_certificate>
sslVersions = tls1.2
requireClientCert = true
sslAltNameToCheck = <identifier of client cert SAN>
sslCommonNameToCheck = <identifier of client cert CN>

Note: Same RPM file was installed on both the Splunk instances. The only difference between the two instances were as follows:

- First instance was configured as an indexer with email alerts.
- Second instance configured as Forwarder that forwards logs to Indexer.

6 Upgrading Splunk

If the update is available, Splunk will display the notification under the "Messages" menu. An administrator can update the Splunk by clicking on this notification which will redirect to the Splunk portal wherein the administrator has to login in the portal with the valid credentials and download the .rpm package. Install this package as root using the system's package manager and verify the update was installed successfully.

Version information in the Splunk can be verified by querying through WEB GUI under the Help-> About tab and similarly the "splunk version" command can be used in the CLI to determine the current installed version on the TOE.

7 Security Relevant Audit Events

The TOE can generate audit records that are stored internally within the TOE whenever an audited event occurs which allows Administrators to discover intentional and unintentional issues with the TOE's configuration and/or operation. Auditing of administrative activities provides information that may be used to hasten corrective action should the system be configured incorrectly. Audit data can also provide an indication of failure of critical portions of the TOE (e.g. a communication channel failure or anomalous activity (e.g. establishment of an administrative session at a suspicious time, repeated failures to establish sessions or authenticate to the TOE) of a suspicious nature.

The TOE generates an audit record whenever an audited event occurs. There are several files the audit system uses to control auditing and write audit records. The fault location for these files is under the directory "opt/splunk/var/log/splunk/"

- **Below sample log shows the successful communication between the Splunk Indexer and SMTP server.**

2020-04-30 06:58:39,424 -0400 INFO sendemail:139 - Sending email. subject="Here is an email notification", results_link="None", recipients="[u'user@mail.splunk.local']", server="mail. splunk.local:465"

- **Below sample log show the successful communication between the indexer and the forwarder.**

04-30-2020 06:52:59.809 -0400 INFO TcpOutputProc - Found currently active indexer. Connected to idx=10.1.3.98:9998, reuse=1.

- **Below sample log shows the certificate validation fails between the Indexer and Forwarder.**

08-20-2020 03:27:33.431 -0400 ERROR X509Verify - X509 certificate (C=US,OU=QA,O=Splunk Inc.,CN=indexer.acumensec.local) failed validation; error=20, reason="unable to get local issuer certificate"
08-20-2020 03:27:33.431 -0400 WARN SSLCommon - Received fatal SSL3 alert. ssl_state='error', alert_description='unknown CA'.
08-20-2020 03:27:33.431 -0400 DEBUG SSLCommon - SESSION ssl=0x7fd83f4b3300 target=? session=(nil) ref=0 ret=-1

08-20-2020 03:27:33.431 -0400 ERROR TcpOutputFd - Connection to host=10.1.3.98:9998 failed. sock_error = 0. SSL Error = error:14090086:SSL routines:ssl3_get_server_certificate:certificate verify failed - please check the output of the `openssl verify` command for the certificates involved; note that if certificate verification is enabled (requireClientCert or sslVerifyServerCert set to "true"), the CA certificate and the server certificate should not have the same Common Name.

- **Below sample log shows the expired certificate between Indexer and Forwarder.**

08-22-2020 04:00:05.478 -0400 ERROR X509Verify - X509 certificate (CN=indexer.acumensec.local,OU=QA,O=Splunk Inc.,C=US) failed validation; error=10, reason="certificate has expired"
08-22-2020 04:00:05.478 -0400 WARN SSLCommon - Received fatal SSL3 alert. ssl_state='error', alert_description='certificate expired'.
08-22-2020 04:00:05.478 -0400 DEBUG SSLCommon - SESSION ssl=0x7f47750b2f80 target=? session=(nil) ref=0 ret=-1
08-22-2020 04:00:05.478 -0400 ERROR TcpOutputFd - Connection to host=10.1.3.98:9998 failed. sock_error = 0. SSL Error = error:14090086:SSL routines:ssl3_get_server_certificate:certificate verify failed - please check the output of the `openssl verify` command for the certificates involved; note that if certificate verification is enabled (requireClientCert or sslVerifyServerCert set to "true"), the CA certificate and the server certificate should not have the same Common Name.

- **Below sample log shows the revoked certificate between Indexer and Forwarder.**

08-20-2020 08:54:21.362 -0400 ERROR X509 - CRL issuer ("/C=US/O=Acumen/OU=CC/CN=Acumen-ICA2") has revoked the certificate with serial 3655436325641345262 (0x32BAB7F888E958EE)
08-20-2020 08:54:21.362 -0400 WARN SSLCommon - Received fatal SSL3 alert. ssl_state='error', alert_description='certificate revoked'.
08-20-2020 08:54:21.362 -0400 DEBUG SSLCommon - SESSION ssl=0x7f95548b2f80 target=? session=(nil) ref=0 ret=-1
08-20-2020 08:54:21.363 -0400 ERROR TcpOutputFd - Connection to host=10.1.3.98:9998 failed. sock_error = 0. SSL Error = error:14090086:SSL routines:ssl3_get_server_certificate:certificate verify failed - please check the output of the `openssl verify` command for the certificates involved; note that if certificate verification is enabled (requireClientCert or sslVerifyServerCert set to "true"), the CA certificate and the server certificate should not have the same Common Name.

8 Uninstalling Splunk Enterprise v8.1

8.1 Uninstalling Process for RHEL7.7

Before uninstalling Splunk make sure to stop the Splunk using the script created in the section 3.3 "stop_splunk.sh" then run the following commands as root.

- To stop Splunk.

  ```
  /home/splunk/stop_splunk.sh
  ```

- To stop dbus, use the script stop_dbus.sh and run the following commands.

  ```
  /home/splunk/stop_dbus.sh
  pkill gnome-keyring
  rm /tmp/dbus-address
  ```

- To clear secret-storage.

  ```
  rm /home/splunk/.gnome2/keyrings/splunk.keyring
  ```

- To completely remove Splunk.

  ```
  yum remove splunk-selinux
  mv /etc/opt/splunk /opt/splunk/etc
  yum remove splunk
  rm -rf /opt/splunk
  ```

8.2 Uninstalling Process for RHEL8.2

Before uninstalling Splunk make sure to stop the Splunk and then issue the following commands as root.

- To stop Splunk

  ```
  runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk stop
  ```

- To Kill the gnome process.

  ```
  pkill gnome-keyring
  rm /tmp/dbus-address
  ```

- To completely remove Splunk.

```
yum remove splunk-selinux
mv /etc/opt/splunk /opt/splunk/etc
yum remove splunk
rm -rf /opt/splunk
```

8.3 Removing Secrets from Secret Store

Below is the command to remove secrets from the secret store.

```
runcon -u system_u -t splunk_t -r system_r splunk secret-storage --remove --no-prompt
<conf-file> <stanza-name> <attribute-name>
```

9 References

The following documents were created and evaluated as part of the Splunk Enterprise CC evaluation:

- Splunk Enterprise v8.1 Security Target - ST version 2.7 dated 01-18-2021
- Splunk Enterprise v8.1 Administrative Guidance for Common Criteria (AGD – this document) - version 0.9 dated 01-20-2021

Below references where used through the lifecycle of the Splunk Enterprise v8.1 evaluation

- To enable the Debug logs in Splunk

https://docs.splunk.com/Documentation/Splunk/8.1.1/Troubleshooting/Enabledebuglogging#log-local.cfg

- To Back up the Splunk configuration

https://docs.splunk.com/Documentation/Splunk/8.1.1/Admin/Backupconfigurations

- Search in Splunk

https://docs.splunk.com/Documentation/Splunk/8.1.1/SearchReference/CLIsearchsyntax

- To configure server.conf

https://docs.splunk.com/Documentation/Splunk/8.1.1/Admin/Serverconf

- To configure web.conf

https://docs.splunk.com/Documentation/Splunk/8.1.1/Admin/Webconf

End of Document