Assurance Activities Report for a Target of Evaluation

Splunk Enterprise 9.0.4

Assurance Activities Report (AAR)
Version 1.0

March 15, 2023

Security Target (Version 1.0)

Evaluated by:

Booz | Allen | Hamilton

Booz Allen Hamilton Common Criteria Test Laboratory
NIAP Lab # 200423
1100 West Street
Laurel, MD 20707

Prepared for:
National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:
Splunk Inc.,
270 Brannan Street
San Francisco, CA 94107

The Author of the Security Target:
Booz Allen Hamilton,
1100 West Street
Laurel, MD 20707

The TOE Evaluation was sponsored by:
Splunk Inc.,
270 Brannan Street
San Francisco, CA 94107

Evaluation Personnel:
Herbert Markle
Christopher Rakaczky
Evan Seiz

Applicable Common Criteria Version

Common Criteria for Information Technology Security Evaluation, April 2017 Version 3.1 Revision 5

Common Evaluation Methodology Version

Common Criteria for Information Technology Security Evaluation, Evaluation Methodology, April 2017 Version 3.1 Revision 5

Table of Contents

1 Purpose

The purpose of this document is to serve as a non-proprietary attestation that this evaluation has satisfied all of the TSS, AGD, and ATE Assurance Activities required by the Protection Profiles/Extended Packages to which the TOE claims exact conformance. This will give system integrators valuable information about product configuration and testing, help to align Common Criteria evaluations with DISA Security Requirements Guides and Security Test Implementation Guides (SRGs/STIGs), and thereby streamline the process for U.S. Government procurement of validated products.

2 TOE Summary Specification Assurance Activities

The evaluation team completed the testing of the Security Target (ST) '*Splunk Enterprise 9.0.4 Security Target, version 1.0*' and confirmed that the TOE Summary Specification (TSS) contains all Assurance Activities as specified by the '*Protection Profile for Application Software, version 1.4*' and '*Functional Package for Transport Layer Security, version 1.1'*. The evaluators were able to individually examine each SFR's TSS statements and determine that they comprised sufficient information to address each SFR claimed by the TOE as well as meet the expectations of the APP PP Assurance Activities.

Through the evaluation of ASE_TSS.1-1, described in the ETR, the evaluators were able to determine that each individual SFR was discussed in sufficient detail in the TSS to describe the SFR being met by the TSF in general. However, in some cases the Assurance Activities that are specified in the claimed source material instruct the evaluator to examine the TSS for a description of specific behavior to ensure that each SFR is described to an appropriate level of detail. The following is a list of each SFR, the TSS Assurance Activities specified for the SFR, and how the TSS meets the Assurance Activities. Additionally, each SFR is accompanied by the source material (AppPP) that defines where the most up-to-date TSS Assurance Activity was defined. Based on the findings this activity is satisfied.

FCS_CKM.1.1 – *"The evaluator shall inspect the application and its developer documentation to determine if the application needs asymmetric key generation services. If not, the evaluator shall verify the generate no asymmetric cryptographic keys selection is present in the ST. Otherwise, the evaluation activities shall be performed as stated in the selection-based requirements."*

Upon inspection of the application and its documentation, it was determined that the application is administratively limited to implement asymmetric key generation services for ECC schemes in support of TLS communications. Section 8.1.1 of the ST states that for TLS communications, the TOE implements the additional key generation functionality in support of ECC schemes using NIST curves P-256, P-384, and P-521 that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4. Based on the findings this activity is satisfied.

FCS_CKM.1.1/AK – *"The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.*

If the application "invokes platform-provided functionality for asymmetric key generation," then the evaluator shall examine the TSS to verify that it describes how the key generation functionality is invoked."

Section 8.1.1 of the ST states that the TOE implements key generation functionality in support of ECC schemes using NIST curves P-256, P-384, and P-521 that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4. The ST does not specify more than one scheme. The TOE uses asymmetric cryptography in support of HTTPS/TLS trusted communications. Based on the findings this activity is satisfied.

FCS_CKM.2.1 – "*The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme."*

Section 8.1.2 of the ST states that the TOE supports Elliptic curve-based key establishment schemes for establishment of HTTPS/TLS communications. Elliptic curve-based key establishment conforms to NIST SP 800-56A. This corresponds with the key generation schemes identified in FCS_CKM.1.1. The ST does not specify more than one scheme. Based on the findings this activity is satisfied.

FCS_COP.1.1/SKC – *This SFR does not contain any AppPP TSS Assurance Activities.*

FCS_COP.1.1/Hash – "*The evaluator shall check that the association of the hash function with other application cryptographic functions (for example, the digital signature verification function) is documented in the TSS."*

Section 8.1.4 of the ST states that the TOE performs cryptographic hashing in support of HTTPS/TLS and that SHA-256, SHA-384, and SHA-512 algorithms are supported. Based on the findings this activity is satisfied.

FCS_COP.1.1/KeyedHash – *This SFR does not contain any AppPP TSS Assurance Activities.*

FCS_COP.1.1/Sig – *This SFR does not contain any AppPP TSS Assurance Activities.*

FCS_HTTPS_EXT.1.1/Client – "*The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818."*

Section 8.17 of the ST details HTTPS/Client and HTTPS/Server behavior and the use of X.509v3 certificates and its reliance of TLS as detailed in the ST. The TOE will reject the connection if the peer certificate presented is invalid or revoked. As this paragraph details both Client and Server interaction with X.509.3 and use of TLS connection there is enough evidence that it complies with RFC 2818.

FCS_HTTPS_EXT.1.2/Client – *This SFR does not contain any AppPP TSS Assurance Activities.*

FCS_HTTPS_EXT.1.3/Client – *This SFR does not contain any AppPP TSS Assurance Activities.*

FCS_HTTPS_EXT.1.1/Server – "*The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818.*"

Section 8.17 of the ST details HTTPS/Client and HTTPS/Server behavior and the use of X.509v3 certificates and its reliance of TLS as detailed in the ST. The TOE will reject the connection if the peer certificate presented is invalid or revoked. As this paragraph details both Client and Server interaction with X.509.3 and use of TLS connection there is enough evidence that it complies with RFC 2818.

FCS_HTTPS_EXT.1.2/Server – *This SFR does not contain any AppPP TSS Assurance Activities.*

FCS_HTTPS_EXT.2.1 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FCS_RBG_EXT.1.1 – "*If "use no DRBG functionality" is selected, the evaluator shall inspect the application and its developer documentation and verify that the application needs no random bit generation services.*

If "implement DRBG functionality" is selected, the evaluator shall ensure that additional FCS_RBG_EXT.2 elements are included in the ST.

If "invoke platform-provided DRBG functionality" is selected, the evaluator performs the following activities. The evaluator shall examine the TSS to confirm that it identifies all functions (as described by the SFRs included in the ST) that obtain random numbers from the platform RBG. The evaluator shall determine that for each of these functions, the TSS states which platform interface (API) is used to obtain the random numbers. The evaluator shall confirm that each of these interfaces corresponds to the acceptable interfaces listed for each platform below.
It should be noted that there is no expectation that the evaluators attempt to confirm that the APIs are being used correctly for the functions identified in the TSS; the activity is to list the used APIs and then do an existence check via decompilation."

The selection for this SFR in the ST is "implement DRBG functionality". As such, the additional FCS_RBG_EXT.2 elements are included in the ST. Based on the findings this activity is satisfied.

FCS_RBG_EXT.2.1 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FCS_RBG_EXT.2.2 – "*Documentation shall be produced - and the evaluator shall perform the activities - in accordance with Appendix C - Entropy Documentation and Assessment and the Clarification to the Entropy Documentation and Assessment Annex."*

A proprietary entropy analyses report was submitted and approved by NIAP as part of the Check-In process. Based on the findings this activity is satisfied.

FCS_STO_EXT.1.1 – "*The evaluator shall check the TSS to ensure that it lists all persistent credentials (secret keys, PKI private keys, or passwords) needed to meet the requirements in the ST. For each of these items, the evaluator shall confirm that the TSS lists for what purpose it is used, and how it is stored."*

The Section 8.1.9 of the ST provides a table of the persistent credentials that the TOE uses that meet the requirements in the ST along with their purpose. The credentials are stored in the GNOME keyring on the underlying platform.
The TOE does not implement functionality to store credentials. The evaluator performed the testing actions as required in the ATE test assurance activity. Based on the findings this activity is satisfied.

FCS_TLS_EXT.1.1 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FCS_TLSC_EXT.1.1 – *"The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component."*

Section 8.1.10 of the ST identifies TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, and TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as the ciphers for TLS v1.2 client support in the

evaluated configuration. These are consistent with the allowed list of ciphers identified in the SFR declaration. Based on the findings this activity is satisfied.

FCS_TLSC_EXT.1.2 – *"The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the application-configured reference identifier, including which types of reference identifiers are supported (e.g. Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported. The evaluator shall ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the product."*

Section 8.1.10 states the reference identifiers are configured within the .conf files to verify Common Name (CN) and/or Subject Alternative Names (SAN) presented identifiers. The CN hostname and SAN hostname (DNS name) are the only supported reference identifiers that can be forced as part of the certificate validation. Additionally it is stated that the TOE does not support the use of URI names, Service names, IP addresses, wildcard certificates, or pinned certificates. Based on the findings this activity is satisfied.

FCS_TLSC_EXT.1.3 – *"If the selection for authorizing override of invalid certificates is made, then the evaluator shall ensure that the TSS includes a description of how and when user or administrator authorization is obtained. The evaluator shall also ensure that the TSS describes any mechanism for storing such authorizations, such that future presentation of such otherwise-invalid certificates permits establishment of a trusted channel without user or administrator action."*

The ST does not support overriding an invalid certificate. Section 8.1.10 states the TOE will not establish a trusted channel if the peer certificate is invalid. Based on the findings this activity is satisfied.

FCS_TLSC_EXT.2.1 – *"The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication. The evaluator shall also ensure that the TSS describes any factors beyond configuration that are necessary in order for the client to engage in mutual authentication using X.509v3 certificates."*

Section 8.1.10 states that the TOE supports mutual TLS authentication using client-side X.509v3 certificates for TLS connections. The TOE will present its client certificate to the server upon request. Additionally, there are no other factors beyond configuration of the specified reference identifiers (CN and SAN) identified. Based on the identification of the CN and SAN being the only supported reference identifiers this work unit is considered satisfied.

FCS_TLSC_EXT.5.1 – *"The evaluator shall verify that TSS describes the Supported Groups Extension."*

Section 8.1.10 states that the TOE supports the use of the Elliptic Curves Extension. The curves presented in the Client Hello include NIST curves secp256r1, secp384r1, and secp521r1.

FCS_TLSS_EXT.1.1 – *"The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component."*

Section 8.1.11 of the ST identifiers TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, and TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as the ciphers for TLS v1.2 client support in the

evaluated configuration. These are consistent with the allowed list of ciphers. Based on the findings this activity is satisfied.

FCS_TLSS_EXT.1.2 – *"The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS versions consistent relative to selections in FCS_TLSS_EXT.1.2."*

Section 8.1.11 of the ST states the TSF will reject any TLS client request that is not using TLS v1.2. Based on the findings this activity is satisfied.

FCS_TLSS_EXT.1.3 – *"The evaluator shall verify that the TSS describes the key agreement parameters of the server's Key Exchange message."*

Section 8.1.11 of the ST states that when acting as a TLS server, the TSF will generate ECDHE over NIST curves secp256r1, secp384r1, and secp521r1 key establishment parameters. Based on the findings this activity is satisfied.

FCS_TLSS_EXT.2.1 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FCS_TLSS_EXT.2.2 – *"The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication."*

Section 8.1.11 of the ST states that the TOE supports mutual TLS authentication using client-side X.509v3 certificates for TLS connections. The TOE will present its server certificate to the client upon request. The TOE will validate the peer certificate used for the connection and will not establish a trusted channel if the peer certificate is invalid. Based on the findings this activity is satisfied.

FCS_TLSS_EXT.2.3 – "*If the product implements mutual authentication, the evaluator shall verify that the TSS describes how the DN and SAN in the certificate is compared to the expected identifier."*

Section 8.1.11 states that the reference identifier can be configured within the .conf files to verify Common Name (CN) and/or Subject Alternative Names (SAN) presented identifiers. The CN hostname and SAN hostname (DNS name) are the only supported reference identifiers that can be forced as part of the certificate validation. Based on the findings this activity is satisfied.

FDP_DAR_EXT.1.1 – "*The evaluator shall examine the TSS to ensure that it describes the sensitive data processed by the application. The evaluator shall then ensure that the following activities cover all of the sensitive data identified in the TSS.*

If not store any sensitive data is selected, the evaluator shall inspect the TSS to ensure that it describes how sensitive data cannot be written to non-volatile memory. The evaluator shall also ensure that this is consistent with the filesystem test below."

The ST declares the "leverage platform-provided functionality to encrypt sensitive data" Section 8.2.1 of the ST describes the use of LUKS and keyrings as the means to store sensitive data. Based on the findings this activity is satisfied.

FDP_DEC_EXT.1.1 – *This SFR does not contain any AppPP TSS Assurance Activities*.

FDP_DEC_EXT.1.2 – *This SFR does not contain any AppPP TSS Assurance Activities*.

FDP_NET_EXT.1.1 - *This SFR does not contain any AppPP TSS Assurance Activities.*

FIA_X509_EXT.1.1 – *"The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm."*

Section 8.3.1 of the ST defines that certificate validation takes place when establishing the following connections:

- TOE (TLS client) to SMTP server via TLS
- TOE (TLS client) to Trusted Data Feed server via HTTPS/TLS
- TOE (TLS server) to Trusted Data Feed client via HTTPS/TLS with mutual authentication

Additionally, the TOE provides an internal mechanism to perform certificate validation and then describes the certificate path algorithm in order for the TOE to validate a certificate. Based on the findings this activity is satisfied.

FIA_X509_EXT.1.2 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FIA_X509_EXT.2 – "*The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates. The evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed."*

Section 8.3.2 of the ST states the TOE uses X.509 certificates for HTTPS/TLS authentication. The use of certificates is enabled by default. However, a security administrator may configure the behavior of this function by specifying whether mutual authentication is supported. The security administrator may also specify the path to a certificate revocation list so that revocation status can be checked during authentication. The actual imported certificates and keys to be used by the TOE are specified through the use of .conf files. While the HTTPS implementation will automatically reject a certificate if it is found to be invalid, a certificate with unknown revocation status is accepted.

FMT_CFG_EXT.1.1 – "*The evaluator shall check the TSS to determine if the application requires any type of credentials and if the application installs with default credentials."*

Section 8.4.1 of the ST states that the TOE requires credentials for remote administration via the web UI and that the initial installation of the TOE prompts the security administrator to create a username and password. There are no default credentials. Based on the findings this activity is satisfied.

FMT_CFG_EXT.1.2 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FMT_MEC_EXT.1.1 – "*The evaluator shall review the TSS to identify the application's configuration options (e.g.settings) and determine whether these are stored and set using the mechanisms supported by the platform or implemented by the application in accordance with the PP-Module for File Encryption. At a minimum the TSS shall list settings related to any SFRs and any settings that are mandated in the operational guidance in response to an SFR.*

Conditional: If "implement functionality to encrypt and store configuration options as defined by "FDP_PRT_EXT.1 in the PP-Module for File Encryption" is selected, the evaluator shall ensure that the TSS identifies those options, as well as indicates where the encrypted representation of these options is stored."

Section 8.4.2 of the ST states that the TOE is capable of using the underlying platform's recommend methods for storing and setting configuration options. In the TOE's evaluated configuration, all configuration information related to the Splunk application is stored in /etc/opt/splunk.

Additionally, the ST states, there are several dedicated configuration files with parameters and settings that are required for CC configuration. These configuration files include: server.conf (back-end communications between splunkd and Splunk Web), web.conf (remote web UI), alert_actions.conf (SMTP), inputs.conf (TLS server for Indexer functionality), outputs.conf (TLS client for Forwarder functionality). These parameters include the ability to configure the cipher suites, set the TLS version for both server and client communication, customize the reference identifier (SAN and CN), identify the X.509 certificate and storage path, enable certificate validation, and enable mutual authentication. See section 7.5 of the Splunk Enterprise 9.0.4 Supplemental Administrative Guidance for Common Criteria for the full description of parameter names and settings related to the SFRs and settings that are mandated for CC compliance. Based on the findings this activity is satisfied.

FMT_SMF.1.1 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FPR_ANO_EXT.1 – *"The evaluator shall inspect the TSS documentation to identify functionality in the application where PII can be transmitted."*

Section 8.5.1 of the ST states that the TOE does not collect personally identifiable information (PII) for security administrators or users. Therefore, there is no case in which the TOE will transmit this data over the network. Based on the findings this activity is satisfied.

FPT_AEX_EXT.1.1 – *"The evaluator shall ensure that the TSS describes the compiler flags used to enable ASLR when the application is compiled."*

Section 8.6.1 of the ST states that the TOE was compiled using the ASLR compilation flags -pie and -fPIE. Based on the findings this activity is satisfied.

FPT_AEX_EXT.1.2 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FPT_AEX_EXT.1.3 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FPT_AEX_EXT.1.4 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FPT_AEX_EXT.1.5 – *This SFR does not contain any AppPP TSS Assurance Activities.*

Section 8.6.1 of the ST states that the TOE was compiled using the -fstack-protector-strong compilation flag. Based on the findings this activity is satisfied.

FPT_API_EXT.1.1 – "*The evaluator shall verify that the TSS lists the platform APIs used in the application."*

Section 8.6.2 of the ST states that Splunk Enterprise ships almost all of the libraries and scripting languages Splunk requires to operate and does not depend on the platform. Scripting languages like Python are part of the TOE and are not platform APIs leveraged by TOE. The only exceptions where Splunk leverages the platform's API (system calls) are listed in Table 13 in the ST.

The evaluator took the system call list from the ST, which was provided by the vendor, and mapped the system calls to the Unix library (.so). The evaluator then mapped the Unix library to the Unix Package that the library is contained. The evaluator then verified that the Unix libraries and packages were installed on the TOE platform. The evaluator was able to verify that the correct packages (or upgraded versions of the packages), and libraries were installed on the evaluated platform. Based on the findings this activity is satisfied.

FPT_IDV_EXT.1.1 – "*If "other version information" is selected the evaluator shall verify that the TSS contains an explanation of the versioning methodology."*

Section 8.6.3 of the ST states that the TOE is versioned with SWID tags that comply with the minimum requirements from ISO/IEC 19770-2:2015. Based on the findings this activity is satisfied.

FPT_LIB_EXT.1.1 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FPT_TUD_EXT.1.1 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FPT_TUD_EXT.1.2 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FPT_TUD_EXT.1.3 – *This SFR does not contain any AppPP TSS Assurance Activities.*

FPT_TUD_EXT.1.4 – "*The evaluator shall verify that the TSS identifies how updates to the application are signed by an authorized source. The definition of an authorized source must be contained in the TSS. The evaluator shall also ensure that the TSS (or the operational guidance) describes how candidate updates are obtained."*

Section 8.6.5 of the ST states that Splunk automatically checks to see if an update is available when a user is authenticated to the web UI. Splunk will notify the authenticated user with a message displayed on the post-authentication page, underneath the "Messages" menu if there is an update available. There is no update message presented to the authenticated user if there is no update available. Splunk does not download updates automatically.

Additionally, the ST states, that after selecting the update URL, the user will be redirected to the authorized Splunk customer portal site where the customer must authenticate prior to being able must manually download the RPM package to the underlying platform. This package must then be manually installed using the platform's RPM application by someone with root privilege. Splunk provides a public key within the RPM and is installed during the initial installation. The root administrator should run the "rpm -K" command which will verify the update against the installed public key prior to installation. The authorized source for the digitally signed updates is "Splunk". Based on the findings this activity is satisfied.

FPT_TUD_EXT.1.5 – *"The evaluator shall verify that the TSS identifies how the application is distributed. If "with the platform" is selected the evaluated shall perform a clean installation or factory reset to confirm that TOE software is included as part of the platform OS. If "as an additional package" is selected the evaluator shall perform the tests in FPT_TUD_EXT.2."*

The "additional package" has been selected for this SFR. Section 8.6.5 states that the package is an RPM and FPT_TUD_EXT.2 is claimed in the ST and the test activities were performed. Based on the findings this activity is satisfied.

FPT_TUD_EXT.2.1 – *This SFR does not contain any AppPP TSS Assurance Activities*

FPT_TUD_EXT.2.2 – *This SFR does not contain any AppPP TSS Assurance Activities*

FTP_DIT_EXT.1.1 – **"***For platform-provided functionality, the evaluator shall verify the TSS contains the calls to the platform that TOE is leveraging to invoke the functionality."*

The ST does not claim "platform-provided functionality". Section 8.7.1 describes that the TOE implements HTTPS/TLS v1.2 for communications. Based on the findings this activity is satisfied.

3 Operational Guidance Assurance Activities

The evaluation team completed the testing of the Operational Guidance, which includes the review of the "*Splunk Enterprise 9.0.4 Supplemental Administrative Guidance for Common Criteria"* (AGD) document and confirmed that the Operational Guidance contains all Assurance Activities as specified by the '*Protection Profile for Application Software, version 1.4*' and '*Functional Package for Transport Layer Security, version 1.1'*. The evaluators reviewed the AppPP to identify the security functionality that must be discussed for the operational guidance. This is prescribed by the Assurance Activities for each SFR and the AGD SARs. The evaluators have listed below each of the SFRs defined in the AppPP that have been claimed by the TOE (some SFRs are conditional or optional) as well as the AGD SAR, along with a discussion of where in the operational guidance the associated Assurance Activities material can be found. The AGD includes references to other guidance documents that must be used to properly install, configure, and operate the TOE in its evaluated configuration. The AGD and its references to other Splunk Enterprise 9.0.4 guidance documents were reviewed to assess the Operational Guidance Assurance Activities. The AGD contains references to these documents in Chapter 4 and these references can also be found below.

The following references are used in this section of the document:

[1] Splunk Enterprise 9.0.4 Supplemental Administrative Guidance for Common Criteria (AGD)

FCS_CKM.1.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_CKM.1.1/AK – "*The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all uses defined in this PP."*

Section 7.4 of the AGD states that the administrator is responsible for performing the operations in Sections 7.5 and 7.6 of the AGD in order to properly configure the TOE such that its cryptographic operations are limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as the TOE already comes pre-configured to meet many of the Common Criteria requirements. Section 7.6 automates many of the remaining configurations through the Common Criteria Mode, and the remaining Section 7.5 have the administrator manually configuring the remaining items (i.e. ciphersuites, algorithms). Based on the findings this activity is satisfied.

FCS_CKM.2.1 – *"The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s)."*

Section 7.4 of the AGD states that the administrator is responsible for performing the operations in Sections 7.5 and 7.6 of the AGD in order to properly configure the TOE such that its cryptographic operations are limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as the TOE already comes pre-configured to meet many of the Common Criteria requirements. Section 7.6 automates many of the remaining configurations through the Common Criteria Mode, and the remaining Section 7.5 have the administrator manually configuring the remaining items (i.e. ciphersuites, algorithms). Based on the findings this activity is satisfied.

FCS_COP.1.1/SKC – "*The evaluator checks the AGD documents to determine that any configuration that is required to be done to configure the functionality for the required modes and key sizes is present."*

Section 7.4 of the AGD states that the administrator is responsible for performing the operations in Sections 7.5 and 7.6 of the AGD in order to properly configure the TOE such that its cryptographic operations are limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as the TOE already comes pre-configured to meet many of the Common Criteria requirements. Section 7.6 automates many of the remaining configurations through the Common Criteria Mode, and the remaining Section 7.5 have the administrator manually configuring the remaining items (i.e. ciphersuites, algorithms). Based on the findings this activity is satisfied.

FCS_COP.1.1/Hash – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_COP.1.1/*KeyedHash* – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_COP.1.1/Sig – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_HTTPS_EXT.1.1/Client – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_HTTPS_EXT.1.2/Client – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_HTTPS_EXT.1.3/Client – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_HTTPS_EXT.1.1/Server – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_HTTPS_EXT.1.2/Server – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_HTTPS_EXT.2.1– *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_RBG_EXT.1.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_RBG_EXT.2.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_RBG_EXT.2.2 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_STO_EXT.1.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_TLS_EXT.1.1 – *The evaluator shall ensure that the selections indicated in the ST are consistent with selections in the dependent components.*

The TLS settings, TLSv 1.2 support, ciphers and ecdhe curves, as defined in Section 7.5.2 of the AGD are all consistent with Section 8.1.10 of the ST. The evaluator has ensured consistency between the AGD and the TSS section.

FCS_TLSC_EXT.1.1 – *"The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the product so that TLS conforms to the description in the TSS."*

Section 7.4 of the AGD states that the administrator is responsible for performing the operations in Sections 7.5 and 7.6 of the AGD in order to properly configure the TOE such that its cryptographic operations are limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as the TOE already comes pre-configured to meet many of the Common Criteria requirements. Section 7.6 automates many of the remaining configurations through the Common Criteria Mode, and the remaining Section 7.5 have the administrator manually configuring the remaining items (i.e. ciphersuites, algorithms).

Specifically, Section 7.5.2 of the AGD provides instructions on how to configure the TOE so that TLS conforms and is consistent with the evaluated configuration as described in the Section 8.1.10 of the ST.

FCS_TLSC_EXT.1.2 – *"The evaluator shall verify that the AGD guidance includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS."*

Section 7.4 of the AGD states that the administrator is responsible for performing the operations in Sections 7.5 and 7.6 of the AGD in order to properly configure the TOE such that its cryptographic operations are limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as the TOE already comes pre-configured to meet many of the Common Criteria requirements. Section 7.6 automates many of the remaining configurations through the Common Criteria Mode, and the remaining Section 7.5 have the administrator manually configuring the remaining items (i.e. ciphersuites, algorithms).

Specifically, Section 7.5.2 of the AGD provides instructions on how to configure the TOE CN and SAN reference identifiers for the purposes of certificate validation in TLS. Based on the findings this activity is satisfied.

FCS_TLSC_EXT.1.3 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FCS_TLSC_EXT.2.1 – *"The evaluator shall ensure that the AGD guidance includes any instructions necessary to configure the TOE to perform mutual authentication. The evaluator also shall verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication."*

Section 7.5.2 of the AGD provides instructions on how to configure the TOE for client side TLS mutual authentication (outputs.conf). Setting the `sslVerifyServerCert` parameter to "true" configures the TOE to validate the server certificate for support of TLS with or without mutual authentication.

FCS_TLSC_EXT.5.1 –*"This SFR does not contain any AppPP AGD Assurance Activities."*

FCS_TLSS_EXT.1.1 – *"The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS."*

Section 7.4 of the AGD states that the administrator is responsible for performing the operations in Sections 7.5 and 7.6 of the AGD in order to properly configure the TOE such that its cryptographic operations are limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as the TOE already comes pre-configured to meet many of the Common Criteria requirements. Section 7.6 automates many of the remaining configurations through the

Common Criteria Mode, and the remaining Section 7.5 have the administrator manually configuring the remaining items (i.e. ciphersuites, algorithms).

Specifically, Section 7.5.2 of the AGD provides instructions on how to configure the TOE CN and SAN reference identifiers for the purposes of certificate validation in TLS. Based on the findings this activity is satisfied.

FCS_TLSS_EXT.1.2 – *"The evaluator shall verify that the AGD guidance includes any configuration necessary to meet this requirement."*

Section 7.4 of the AGD states that the administrator is responsible for performing the operations in Sections 7.5 and 7.6 of the AGD in order to properly configure the TOE such that its cryptographic operations are limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as the TOE already comes pre-configured to meet many of the Common Criteria requirements. Section 7.6 automates many of the remaining configurations through the Common Criteria Mode, and the remaining Section 7.5 have the administrator manually configuring the remaining items (i.e. ciphersuites, algorithms).

Specifically, Section 7.5.2 of the AGD provides instructions on how to configure the TOE CN and SAN reference identifiers for the purposes of certificate validation in TLS. Based on the findings this activity is satisfied.

FCS_TLSS_EXT.1.3 – *"The evaluator shall verify that any configuration guidance necessary to meet the requirement must be contained in the AGD guidance."*

Section 7.4 of the AGD states that the administrator is responsible for performing the operations in Sections 7.5 and 7.6 of the AGD in order to properly configure the TOE such that its cryptographic operations are limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as the TOE already comes pre-configured to meet many of the Common Criteria requirements. Section 7.6 automates many of the remaining configurations through the Common Criteria Mode, and the remaining Section 7.5 have the administrator manually configuring the remaining items (i.e. ciphersuites, algorithms).

Specifically, Section 7.5.2 of the AGD provides instructions on how to configure the TOE CN and SAN reference identifiers for the purposes of certificate validation in TLS. Based on the findings this activity is satisfied.

FCS_TLSS_EXT.2.1 – This SFR does not contain any AppPP AGD Assurance Activities.

FCS_TLSS_EXT.2.2 – *"The evaluator shall verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication. The evaluator shall ensure that the AGD guidance includes instructions for configuring the server to require mutual authentication of clients using these certificates."*

Section 7.5.2 of the AGD provides instructions on how to configure the TOE server side TLS mutual authentication (inputs.conf). Setting the `requireClientCert` parameter to "true" configures the TOE to validate request and validate the client certificate for support of TLS with mutual authentication.

FCS_TLSS_EXT.2.3 – *"If the DN is not compared automatically to the domain name, IP address, username, or email address, the evaluator shall ensure that the AGD guidance includes configuration of the expected identifier or the directory server for the connection."*

Section 7.4 of the AGD states that the administrator is responsible for performing the operations in Sections 7.5 and 7.6 of the AGD in order to properly configure the TOE such that its cryptographic operations are limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as the TOE already comes pre-configured to meet many of the

Common Criteria requirements. Section 7.6 automates many of the remaining configurations through the Common Criteria Mode, and the remaining Section 7.5 have the administrator manually configuring the remaining items (i.e. ciphersuites, algorithms).

Specifically, Section 7.5.2 of the AGD provides instructions on how to configure the TOE CN and SAN reference identifiers for the purposes of certificate validation in TLS. Based on the findings this activity is satisfied.

FDP_DAR_EXT.1.1 - *This SFR does not contain any AppPP AGD Assurance Activities.*

FDP_DEC_EXT.1.1 – *"The evaluator shall perform the platform-specific actions below and inspect user documentation to determine the application's access to hardware resources. The evaluator shall ensure that this is consistent with the selections indicated. The evaluator shall review documentation provided by the application developer and for each resource which it accesses, identify the justification as to why access is required"*

Section 5.2, Table 1 of the AGD states that for the Host Platform component, the TOE requires network resources from the host platform. Based on the findings this activity is satisfied.

FDP_DEC_EXT.1.2 – *"The evaluator shall perform the platform-specific actions below and inspect user documentation to determine the application's access to sensitive information repositories. The evaluator shall ensure that this is consistent with the selections indicated. The evaluator shall review documentation provided by the application developer and for each sensitive information repository which it accesses, identify the justification as to why access is required."*

The AGD makes no claims for sensitive repositories which is consistent with the selection in the ST. Based on the findings this activity is satisfied.

FDP_NET_EXT.1 - *This SFR does not contain any AppPP AGD Assurance Activities.*

FIA_X509_EXT.1.1 – This SFR does not contain any AppPP AGD Assurance Activities.

FIA_X509_EXT.1.2 – This SFR does not contain any AppPP AGD Assurance Activities.

FIA_X509_EXT.2 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FMT_CFG_EXT.1.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FMT_CFG_EXT.1.2 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FMT_MEC_EXT.1.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FMT_SMF.1.1 – *"The evaluator shall verify that every management function mandated by the PP is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function."*

Section 7.5.2 of the AGD describes how to enable/disable supported TLS cipher suites via the *.conf files within the application configuration files directory. Section 7.12 of the AGD describes how to query the current version of the TOE via the UI and CLI. Based on the findings this activity is satisfied.

FPR_ANO_EXT.1.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_AEX_EXT.1.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_AEX_EXT.1.2 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_AEX_EXT.1.3 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_AEX_EXT.1.4 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_AEX_EXT.1.5 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_API_EXT.1.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_LIB_EXT.1.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_TUD_EXT.1.1 – *"The evaluator shall check to ensure the guidance includes a description of how updates are performed."*

Section 7.12 of the AGD describes the procedures for updating the TOE. Based on the findings this activity is satisfied.

FPT_TUD_EXT.1.2 – *"The evaluator shall verify guidance includes a description of how to query the current version of the application."*

Section 7.12 of the AGD describes how to query the current version of the TOE via the UI and CLI. Based on the findings this activity is satisfied.

FPT_TUD_EXT.1.3 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_TUD_EXT.1.4 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_TUD_EXT.1.5 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_TUD_EXT.2.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FPT_TUD_EXT.2.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

FTP_DIT_EXT.1.1 – *This SFR does not contain any AppPP AGD Assurance Activities.*

4 Security Assurance Requirements

This section addresses assurance activities that are defined in the *Protection Profile for Application Software* [AppPP] that correspond with Security Assurance Requirements.

AGD_OPE.1 – "*Some of the contents of the operational guidance will be verified by the evaluation activities in Section 5.1 Security Functional Requirements and evaluation of the TOE according to the [CEM]. The following additional information is also required."*

If cryptographic functions are provided by the TOE, the operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE. The documentation must describe the process for verifying updates to the TOE by verifying a digital signature – this may be done by the TOE or the underlying platform.

The evaluator shall verify that this process includes the following steps:

- *Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).*

- *Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the digital signature. The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.*

  Section 7.4 of the AGD states that the administrator is responsible for performing the operations in Sections 7.5 and 7.6 of the AGD in order to properly configure the TOE such that its cryptographic operations are limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as the TOE already comes pre-configured to meet many of the Common Criteria requirements. Section 7.6 automates many of the remaining configurations through the Common Criteria Mode, and the remaining Section 7.5 have the administrator manually configuring the remaining items (i.e. ciphersuites, algorithms).

  Specifically, Section 7.5.2 of the AGD provides instructions on how to configure the TOE so that TLS conforms to the evaluated configuration as described in the TSS in the ST.

  Section 7.4 of the AGD states that "The use of other cryptographic engines and cryptographic settings were not evaluated nor tested during the Common Criteria evaluation of the TOE."

  Instructions for obtaining and staging the update itself are outlined in Section 7.12 of the AGD. This section also describes how to initiate the update process using the platform's "rpm" application and to verify the update using a digital signature with the "rpm -K <filename.rpm>" command and verify that the update was successful by re-querying the version after installation.

  The AGD makes it clear in Section 2 that "any functionality that is not described here or in the Splunk Enterprise 9.0.4. Security Target was not evaluated and should be exercised at the user's risk."

AGD_PRE.1 – "*As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST."*

Section 5.1 of the AGD describes the TOE components in the evaluated configuration: The TOE is the Splunk Enterprise 9.0.4 ("Splunk") application executing on a Linux OS. In the evaluated configuration, Splunk Enterprise 9.0.4 is installed on top of the RHEL OS and is configured as an Indexer. Section 5.3 of the AGD contains instructions for the Security Administrator to ensure that the operational environment will fulfill its role in supporting the TOE. These instructions match the assumptions for the TOE's operational environment in Section 4.3 of the ST.

ALC_CMC.1 – "*The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product."*

The ST clearly and consistently states the version as Splunk Enterprise 9.0.4.

The AGD documents clearly indicate the version as Splunk Enterprise 9.0.4

Splunk's website and support site clearly delineates between different versions for both obtaining the product download as well as for online documentation help where one needs to select the correct version.

ALC_CMS.1 – *"The " The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the evaluation activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component. Life-cycle support is targeted aspects of the developer's life-cycle and instructions to providers of applications for the developer's devices, rather than an in-depth examination of the TSF manufacturer's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation.*

The evaluator shall ensure that the developer has identified (in guidance documentation for application developers concerning the targeted platform) one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the developer shall provide information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are invoked (e.g., compiler flags). The evaluator shall ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled. The evaluator shall ensure that the TSF is uniquely identified (with respect to other products from the TSF vendor), and that documentation provided by the developer in association with the requirements in the ST is associated with the TSF using this unique identification."

Splunk has a whole online documentation line that supports the development of Apps for each of the different versions including the specific version of the TOE 9.0.4 (https://dev.splunk.com/enterprise/). Splunk (the TOE) is the development framework for building apps for the TOE. The TOE provides the libraries for app development, structure requirements, and integration requirements. The developer creates an app in the Splunk Web (App) Framework and supports only Simple XML, Simple XML jS/CSS extensions, HTML. The code can be created outside of the framework but must be imported as part of the integration process. The overflow protection is automatic with Splunk framework as it is compiled with the –fstack-protect-strong compiler flag as documented in the ST. Additionally, the OS should be configured per Splunk® Enterprise 9.0.4 Supplemental Administrative Guidance for Common Criteria.

ALC_TSU_EXT.1 – "*The evaluator shall verify that the TSS contains a description of the timely security update process used by the developer to create and deploy security updates. The evaluator shall verify that this description addresses the entire application. The evaluator shall also verify that, in addition to the TOE developer's process, any third-party processes are also addressed in the description. The evaluator shall also verify that each mechanism for deployment of security updates is described.*

The evaluator shall verify that, for each deployment mechanism described for the update process, the TSS lists a time between public disclosure of a vulnerability and public availability of the security update to the TOE patching this vulnerability, to include any third-party or carrier delays in deployment. The evaluator shall verify that this time is expressed in a number or range of days.

The evaluator shall verify that this description includes the publicly available mechanisms (including either an email address or website) for reporting security issues related to the TOE. The evaluator shall verify that the description of this mechanism includes a method for protecting the report either using a public key for encrypting email or a trusted channel for a website."

Section 8.6.5.1 of the ST identifies the developer policy on Timely Security Updates that fulfills the requirement. Any feedback that necessitates a fix will result in a patch to Splunk itself so there is no third-

party update process to consider when updating the TOE. Any security fixes will be released as new packages in the same manner as any feature. Any implementation flaws are expected to be addressed within 90 days of reporting. This process was verified during the course of the evaluation

5 Test Assurance Activities (Test Report)

The following sections demonstrate that all ATE Assurance Activities for the TOE have been met. This evidence has been presented in a manner that is consistent with the "Reporting for Evaluations Against NIAP-Approved Protection Profiles" guidance that has been provided by NIAP. Specific test steps and associated detailed results are not included in this report in order for it to remain non-proprietary. The test report is a summarized version of the test activities that were performed as part of creating the Evaluation Technical Report (ETR).

5.1 Platforms Tested and Composition

The TOE was tested in the Booz Allen Hamilton facility in Laurel, Maryland from October 2022 to March 2023. The evaluation team set up a test environment for the independent functional testing that allowed them to perform the assurance activities against the TOE over the SFR relevant interfaces. The evaluation team performed testing on both the command line and graphical user administrative interfaces.

Splunk Enterprise 9.0.4 is a software-only TOE. All hardware that is present is part of the TOE's Operational Environment. The following system configurations were used for the testing of the TOE (both Indexer and Forwarder for each):

- Configuration 1 (referred to as C1 in figures below):
  - Red Hat Enterprise Linux 8.2 64 bit
  - Intel(R) Xeon(R) CPU E5-2630v4
  - 16 GB RAM
  - 500 GB disk
- Configuration 2 (referred to as C2 in figures below):
  - Red Hat Enterprise Linux 7.9 64 bit
  - Intel(R) Xeon(R) CPU E5-2630v4
  - 16 GB RAM
  - 500 GB disk

Testing could have been accomplished on one machine by reconfiguring the same device from an Indexer to a Forwarder. Two machines are being used to avoid introducing errors that could be caused from reconfiguring the platforms multiple times after the start of testing. Additionally, this method also provides clarity of evidence so that when the evaluator sees a certain IP address it is known what platform it is and that it is consistently performing the same TLS client or TLS server function.

5.1.1 Test Configuration

The following was methodology was applied to each of the RHEL 8.2 and 7.9 based configuration environments:

Configuration 1 refers to the TOE installed on RHEL 8.2 platforms.*
Configuration 2 refers to the TOE installed on RHEL 7.9 platforms.*
Full testing was performed on both RHEL environment.

The TOE platform is configured with the Indexer functionality (C1 in the figures below):

- 1st device with full instance of Splunk Enterprise 9.0.4 software installed*
- TOE's Remote administration Web UI (web.conf) capabilities perform the HTTPS/TLS server function. (no mutual auth) [E2]
- TOE's notifications capability (alert_action.conf) acts as a TLS client for connection to the

SMTP server. (no mutual auth) [E3]

- The TOE's Indexer capability (inputs.conf) performs the HTTP/TLS server functions when establishing a connection to a Trusted Data Feed (client). (w/ mutual auth) [E4]
- Referred to as the TOE=Indexer

The OE Trusted Data Feed platform (C2 in the figures below): :

- 2nd device with full instance of Splunk Enterprise 9.0.4 software installed*
- Splunk's Forwarder capability (output.conf) performs the HTTPS/TLS client functions when establishing the connection to a Trusted Data Feed (server). (w/ mutual auth) [E4]
- Provides an easily identifiable IP address for testing Client functionality vs Server functionality.
- Referred to as OE Trusted Data Feed (client)
- tcpdump 4.9.2

*Same rpm was used to fully install Splunk Enterprise 9.0.4 on all 4 instances (two RHEL 8.2 and two RHEL 7.9)

The 1st device was considered the main TOE platform for testing. The one exception is when testing involved analyzing the FCS_HTTPS_EXT.1/Client, FCS_TLSC_EXT.1, and FCS_TLSC_EXT.2 with mutual authentication between the TOE and Trusted Data Feed [E5] functionality. As this functionality can only be provided by the TOE being configured with the Forwarder functionality (output.conf).

- SMTP Server (C3)
  - Debian GNU/Linux 11 (bullseye)
  - Postfix version 3.5.13
- CRL distribution point (C4)
  - Linux catlsvcs 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64 GNU/Linux
- 2 Management Workstations (C5 and C6)
  - Debian GNU/Linux 11
  - OpenSSL 1.1.1n 15 Mar 2022 (Library: OpenSSL 1.1.1k 25 Mar 2021)
  - OpenSSL 1.0.2u 20 Dec 2019
  - Chromium (version 110.0.5481.78 (Official Build) (64-bit))

The following test tools were installed on multiple test workstations and servers for testing purposes:

- ClamAV version 0.105.1 loaded on the TOE platform:
- Burp Suite Pro (version 2023.1.2)
- Test Machine (Linux kali 4.19.0-kali4-amd64 #1 SMP Debian 4.19.28-2kali1 (2019-03-18) x86_64 GNU/Linux)
  - Man-in-the-Middle (MITM) Packet Modification Tool
- Wireshark version 3.6.7
- tcpdump 4.9.2

Interface declarations:

- E1: Local administrator to TOE via Splunk CLI – This is the local interface that a local administrator uses to securely manage the TOE.

- E2: Remote administrator to TOE via Web UI – This is the remote interface in which the remote administrator uses to securely manage the TOE. TOE acts as HTTPS/TLS server (no mutual authentication) for the secure communications.
- E3: TOE Indexer to SMTP server – This interface is used by the TOE to send out configured alerts. TOE acts as a TLS client (no mutual authentication) for the secure communications.
- E4: External Trusted Data Feed to TOE Indexer – This interface is used by the TOE to the receive data from the Trusted Data Feed client. TOE acts as a HTTPS/TLS server with mutual authentication.
- E5: TOE Forwarder to External Trusted Data Feed receiver– This interface is used by the TOE to the transmit data to the Trusted Data Feed server. TOE acts as a HTTPS/TLS client with mutual authentication.

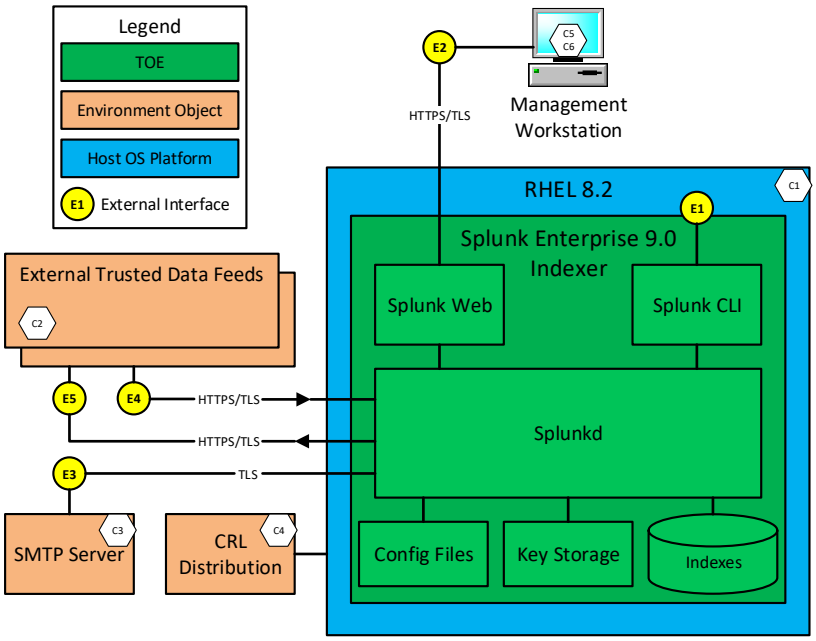

Figure 1 – Test Environment 1

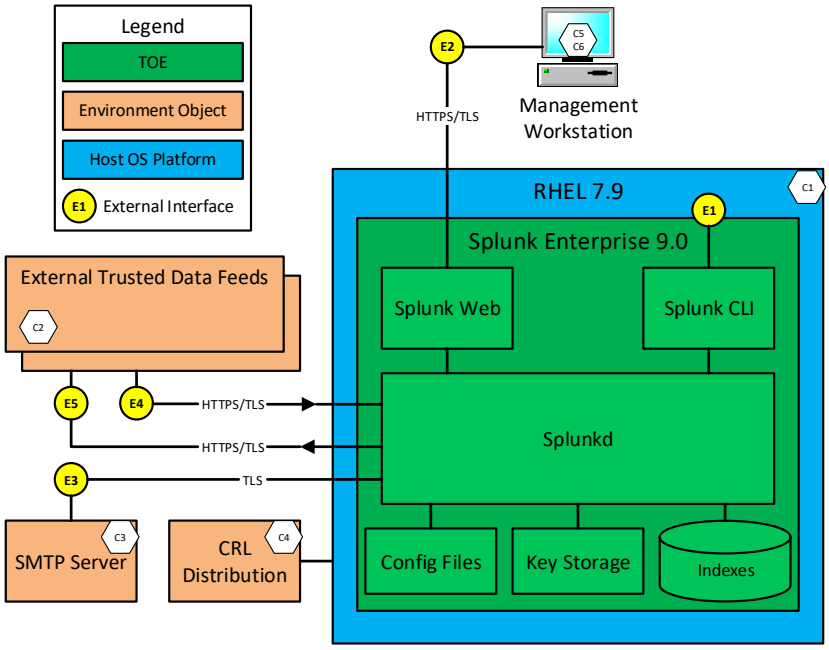

Figure 2 – Test Environment 2

5.2 Omission Justification

There is no omission justification required. All TOE instances were installed using the same installation executable. Each RHEL environment is being fully tested. All external interfaces are being exercised.

5.3 Test Cases

The evaluation team completed the functional testing activities within Booz Allen Hamilton's test environment. The evaluation team conducted a set of testing that includes all ATE Assurance Activities as specified by the '*Protection Profile for Application Software, version 1.4*' and '*Functional Package for Transport Layer Security, version 1.1'*. The evaluators reviewed the AppPP to identify the security functionality that must be verified through functional testing. This is prescribed by the Assurance Activities for each SFR.

Note that some SFRs do not have test Assurance Activities associated with them at the element level (e.g. FCS_CKM.1.1). In such cases, testing for the SFR is considered to be satisfied by completion of all Assurance Activities at the component level.

The following lists for each ATE Assurance Activity, the test objective, test instructions, test steps, and test results. Note that unless otherwise specified, the test configuration is to be in the evaluated configuration as defined by the OPE. For example, some tests require the TOE to be brought out of the evaluated configuration to temporarily disable cryptography to prove that the context of transmitted data is accurate. As part of the cleanup for each test, the TOE is returned to the evaluated configuration.

5.3.1 Cryptographic Support

Test cases for FCS_CKM.1/AK, FCS_CKM.2, FCS_COP.1/SKC, FCS_COP.1/Hash, FCS_COP.1/SIG, FCS_COP.1/KeyedHash, and FCS_RBG_EXT.2 are not included within this section. This is because the ATE Assurance Activities have been satisfied by the vendor having the TOE's algorithms assessed under Cryptographic Algorithm Validation Program (CAVP). As part of CAVP validation the TOE's

cryptographic algorithms went through CAVS testing which directly maps to these SFRs' ATE Assurance Activities. Below is the results of the CAVP validation for the certificates:

SFR	Cert Name (Claimed Algorithm)	CAVP Cert. #
FCS_CKM.1 Key generation	ECDSA: 186-4 Key Pair Generation and Private Key Validation (P-256, P-384, P-521)	A2913
FCS_CKM.1/AK Asymmetric key generation	ECDSA: 186-4 Key Pair Generation and Private Key Validation (P-256, P-384, P-521)	A2913
FCS_CKM.2 Key establishment	ECDHE: KAS-ECC (P-256, P-384, P-521)	A2913
FCS_COP.1/SKC Encryption/decryption	AES (CBC-256, GCM-128, GCM-256, CTR-256)	A2913
FCS_COP.1/Hash Hash	SHS (SHA-256, SHA-384, SHA-512)	A2913
FCS_COP.1/Sig Signing and verification	ECDSA: Signature Generation and Signature Verification (P-256: SHA-256, SHA-384, SHA512 P-384: SHA-256, SHA-384, SHA512 P-521: SHA-256, SHA-384, SHA512)	A2913
FCS_COP.1/KeyedHash Keyed-hash message authentication	HMAC (HMAC-SHA-256, HMAC-SHA-384)	A2913
FCS_RBG_EXT.1 Random Bit Generation	DRBG (CTR-DRBG) requires AES-CTR 256 bit	A2913

001	**FCS_HTTPS_EXT.1.1/Client – HTTPS Protocol**
Test Objective:	The evaluator shall attempt to establish an HTTPS connection with a webserver, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS.
1. Begin capturing packets. 2. Establish a HTTPS connection from the TOE (Forwarder) to the external Trusted Data Feed (Indexer): runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk stop runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk start 3. Stop capturing packets. 4. Inspect the captured packets and verify that the connection succeeded and that the traffic is identified as TLS or HTTPS.	
Test Results:	The evaluator established a HTTPS connection from the Splunk Forwarder to the Splunk Indexer, captured packets, and observed that the connection was successful and that the traffic was identified as TLS / HTTPS.
Execution Method:	Manual

002	**FCS_HTTPS_EXT.1.2/Client – HTTPS Protocol**
Test Objective:	Other tests are performed in conjunction with the TLS package.
Testing performed in FCS_TLSC_EXT.1 satisfies the testing that is required in this test Assurance Activity.	
Test Results:	Testing performed in FCS_TLSC_EXT.1 satisfies the testing that is required in this test Assurance Activity.

Execution Method:	N/A

003	**FCS_HTTPS_EXT.1.3/Client – HTTPS Protocol**
Test Objective:	Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1, and the evaluator shall perform the following test: Test 1: The evaluator shall demonstrate that using a certificate without a valid certification path results in the selected action in the SFR. If "notify the user" is selected in the SFR, then the evaluator shall also determine that the user is notified of the certificate validation failure. Using the administrative guidance, the evaluator shall then load a certificate or certificates to the Trust Anchor Database needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that again, using a certificate without a valid certification path results in the selected action in the SFR, and if "notify the user" was selected in the SFR, the user is notified of the validation failure.
1. Configure the client's trust anchor database to not have a certificate needed to validate the presented certificate. 2. Begin capturing packets between the TOE and the client. 3. Initiate a connection from the client to the TOE. 4. Stop capturing packets between the TOE and the client. 5. Verify that the attempted connection is denied. 6. Load the certificate needed to validate the client certificate presented in Step 3 to the TOE's Trust Anchor Database. 7. Repeat Steps 2 through 4. 8. Verify that the connection is successful.	
Test Results:	The evaluator demonstrated that when the client's trust anchor database contained the required certificates to validate presented certificates, the connection was successful. In the case when the client's trust anchor database did not contain the required certificates to validated presented certificates, the connection was unsuccessful. This was confirmed by examining packets that were captured during connection attempts from the Splunk Forwarder to the Splunk Indexer.
Execution Method:	Manual

004	**FCS_HTTPS_EXT.1.1/Server – HTTPS Protocol**
Test Objective:	The evaluator shall attempt to establish an HTTPS connection to the TOE using a client, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS.
Prepare the environment for capturing packets between the management workstation and the TOE (Indexer). 1. Begin capturing packets between the TOE webserver and the Test Workstation. 2. On the Test Workstation, establish a connection to https://<TOE_IP_ADDRESS>:8000 3. Stop capturing packets between the TOE webserver and the Test Workstation. 4. Inspect the captured packets and verify that the connection succeeded and that the traffic is identified as TLS or HTTPS. Prepare the environment for capturing packets between the TOE (Indexer) and the external Trusted Data Feed (Forwarder). 1. Begin capturing packets. 2. Stop the Trusted Data Feed (Forwarder). 3. Start the Trusted Data Feed (Forwarder). 4. Stop capturing packets. 5. Inspect the captured packets and verify that the connection succeeded and that the traffic is identified as TLS or HTTPS.	
Test Results:	The evaluator verified that the connection between the Test Workstation and the Splunk Indexer web server was successful and that it was identified as TLS / HTTPS. The evaluator then verified that connection between the Splunk Forwarder and Splunk Indexer was successful and that it was identified as TLS / HTTPS.

Execution Method:	Manual

005	**FCS_HTTPS_EXT.1.2/Server – HTTPS Protocol**
Test Objective:	Other tests are performed in conjunction with the TLS package.
Other tests are performed in conjunction with the TLS package.	
Test Results:	Other tests are performed in conjunction with the TLS package.
Execution Method:	N/A

006	**FCS_HTTPS_EXT.2.1/Server – HTTPS Protocol**
Test Objective:	Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1, and the evaluator shall perform the following test: Test 1: The evaluator shall demonstrate that using a certificate without a valid certification path results in the selected action in the SFR. Using the administrative guidance, the evaluator shall then load a certificate or certificates to the Trust Anchor Database needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that again, using a certificate without a valid certification path results in the selected action in the SFR.
This is tested in FIA_X509_EXT.1.1 – Test 1 (Test Case 066).	
Test Results:	**This is tested in FIA_X509_EXT.1.1 – Test 1 (Test Case 066).**
Execution Method:	N/A

000	**FCS_RBG_EXT.1.1 - Random Bit Generation Services**
Test Objective:	If "invoke platform-provided DRBG functionality" is selected, the following tests shall be performed: The evaluator shall decompile the application binary using a decompiler suitable for the application (TOE). The evaluator shall search the output of the decompiler to determine that, for each API listed in the TSS, that API appears in the output. If the representation of the API does not correspond directly to the strings in the following list, the evaluator shall provide a mapping from the decompiled text to its corresponding API, with a description of why the API text does not directly correspond to the decompiled text and justification that the decompiled text corresponds to the associated API. The following are the per-platform list of acceptable APIs: The evaluator shall verify that the application collects random from /dev/random or /dev/urandom.
N/A – The ST states that RBG is implemented. Therefore, this test doesn't apply.	
Test Results:	N/A
Execution Method:	N/A

007	**FCS_STO_EXT.1.1 – Storage of Secrets**
Test Objective:	For all credentials for which the application implements functionality, the evaluator shall verify credentials are encrypted according to FCS_COP.1/SKC or conditioned according to FCS_CKM.1.1/AK and FCS_CKM.1/PBKDF. For all credentials for which the application invokes platform-provided functionality, the evaluator shall perform the following actions which vary per platform. For Linux: The evaluator shall verify that all keys are stored using Linux keyrings.
1. As the root user on the TOE platform, remove the Linux keyring files: rm /home/splunk/.local/share/keyrings/*	

2. Switch to the splunk user context and initialize dbus-run-session:

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

3. Confirm the dbus process is running:

   ps -auxZ | grep dbus

4. Initialize the GNOME keyring daemon with the "--unlock" option, specify a password at the input, then using the keyboard, press [ENTER] followed by [CTRL] + [D]:

   gnome-keyring-daemon --unlock
   password123![ENTER]
   [CTRL] + [D]

5. Output the keys stored in the keyring's initialized state using the "splunk secret-storage" tool:

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

6. Insert all defined keys in the Security Target TSS for this SFR into the keyring:

**
both Indexer and Forwarder
**

**
required (audit, distsearch):

runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk secret-storage --write --no-prompt distsearch tokenExchKeys privateKeyPassphrase password4dist

runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk secret-storage --write --no-prompt audit auditTrail privatekeyPassphrase password4audit

**

runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk secret-storage --write --no-prompt server sslConfig sslPassword P@ssw0rd!

runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk secret-storage --write --no-prompt server kvstore sslPassword P@ssw0rd!

runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk secret-storage --write --no-prompt web settings sslPassword P@ssw0rd!

**
Forwarder
**

runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk secret-storage --write --no-prompt outputs tcpout sslPassword P@ssw0rd!

***
Indexer
**

runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk secret-storage --write --no-prompt alert_actions email auth_password password4SMTP

runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk secret-storage --write --no-prompt inputs SSL

sslPassword P@ssw0rd!

7. Output the keys stored in the keyring's current state using the "splunk secret-storage" tool:

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

8. Output the key values stored in the keyring's current state using the "secret-tool" tool:

   secret-tool lookup splunk store

9. Ensure that for each key / credential defined in the Security Target TSS for this SFR, there is a corresponding match in the output from the previous step.

Test Results:	The evaluator confirmed that all relevant keys are stored using Linux keyrings by removing any existing keyrings, initializing a new keyring instance, outputting the data in the keyring before inserting keys, then inserting keys, and then finally outputting the data in the keyring to confirm the keys written to the keyring by the application were stored in the keyring.
Execution Method:	Manual

009	**FCS_TLSC_EXT.1.1 – TLS Client Protocol**
Test Objective:	The evaluator shall also perform the following tests: Test 1: The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

1. Begin capturing packets between the TOE and the remote TLS server.
2. Configure the TOE to only use the TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ciphersuite.
3. Stimulate TOE to force a connection to the remote TLS server.
4. Verify that the ciphersuite specified in Step 1 was used and that the connection was successful.
5. Repeat Steps 2-4, except, replace TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 with TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256.
6. Repeat Steps 2-4, except, replace TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 with TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384.
7. Stop capturing packets between the TOE and the remote TLS server.

Test Results:	The evaluator confirmed that each of the claimed cipher suites were able to successfully be used to establish a TLS connection by examining the TLS handshake. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

010	**FCS_TLSC_EXT.1.1 – TLS Client Protocol**
Test Objective:	Test 2: The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation. The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and verify that a connection is established. The evaluator shall repeat this test using a different, but otherwise valid and trusted, certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension and ensure that a connection is not established. Ideally, the two certificates should be similar in structure, the types of identifiers used, and the chain of trust.

Valid Certificate

1. Load the certificate containing the Server Authentication purpose onto the remote TLS server.
2. Begin capturing packets between the TOE and the remote TLS server.
3. Stimulate TOE to force a connection to the remote TLS server.

4. Stop capturing packets between the TOE and the remote TLS server. 5. Analyze packet capture to verify that the server presented the certificate containing the Server Authentication purpose and that the connection was successful. **Invalid Certificate** 1. Load the certificate lacking the Server Authentication purpose onto the remote TLS server. 2. Begin capturing packets between the TOE and the remote TLS server. 3. Stimulate TOE to force a connection to the remote TLS server. 4. Stop capturing packets between the TOE and the remote TLS server. 5. Analyze packet capture to verify that the server presented the certificate lacking the Server Authentication purpose and that the connection was unsuccessful.	
Test Results:	The evaluator demonstrated that when a presented certificate contained the Server Authentication purpose, the TLS connection was successful. When the presented certificate lacked the Server Authentication purpose, the TLS connection was unsuccessful. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

011	**FCS_TLSC_EXT.1.1 – TLS Client Protocol**
Test Objective:	The evaluator shall also perform the following tests: Test 3: The evaluator shall send a server certificate in the TLS connection that does not match the server-selected cipher suite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA certificate while using one of the ECDSA cipher suites.) The evaluator shall verify that the product disconnects after receiving the server's Certificate handshake message.
1. Configure the server to use an RSA certificate. 2. Configure the server to use an RSA ciphersuite. 3. Begin capturing packets between the TOE and the remote server. 4. Execute the command to perform the man-in-the-middle modification (MITM) to the Client Hello and Server Hello such that the "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256" ciphersuite is sent. 5. Cause the TOE to establish a TLS connection to the remote server. 6. Stop capturing packets between the TOE and the remote server. 7. Inspect the packet capture to verify that a TLS connection could not be established, and that the client disconnected after receiving the Server's Certificate handshake message.	
Test Results:	The evaluator demonstrated that the TLS connection is unsuccessful after the TOE receives a RSA certificate, but the server selected ciphersuite (TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256) is incompatible with the certificate type. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

012	**FCS_TLSC_EXT.1.1 – TLS Client Protocol**
Test Objective:	The evaluator shall also perform the following tests: Test 4: The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the client denies the connection.
1. Configure the remote server to use an ECDSA certificate. 2. Configure the remote server to the TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ciphersuite. 3. Begin capturing packets between the TOE and the remote server. 4. Execute the command to perform the man-in-the-middle modification (MITM) to the Server Hello such that the "TLS_NULL_WITH_NULL_NULL" ciphersuite is sent. 5. Cause the TOE to establish a TLS connection to the remote server. 6. Stop capturing packets between the TOE and the remote server. 7. Inspect the packet capture to verify that a TLS connection could not be established, and that the client denies the connection after receiving the Server Hello message.	
Test Results:	The evaluator demonstrated that the TLS connection is unsuccessful after the TOE receives a

	Server Hello with a cipher suite selection of "TLS_NULL_WITH_NULL_NULL". This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

013	**FCS_TLSC_EXT.1.1 – TLS Client Protocol**
Test Objective:	Test 5: The evaluator shall perform the following modifications to the traffic: Test 5.1: Change the TLS version selected by the server in the Server Hello to an undefined TLS version (for example 1.5 represented by the two bytes 03 06) and verify that the client rejects the connection. Test 5.2: Change the TLS version selected by the server in the Server Hello to the most recent unsupported TLS version (for example 1.1 represented by the two bytes 03 02) and verify that the client rejects the connection. Test 5.3: [conditional] If DHE or ECDHE cipher suites are supported, modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client does not complete the handshake and no application data flows. Test 5.4: Modify the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message. The evaluator shall verify that the client does not complete the handshake and no application data flows. Test 5.5: [conditional] If DHE or ECDHE cipher suites are supported, modify the signature block in the server's Key Exchange handshake message, and verify that the client does not complete the handshake and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted. Test 5.6: Modify a byte in the Server Finished handshake message, and verify that the client does not complete the handshake and no application data flows. Test 5.7: Send a message consisting of random bytes from the server after the server has issued the Change Cipher Spec message and verify that the client does not complete the handshake and no application data flows. The message must still have a valid 5-byte record header in order to ensure the message will be parsed as TLS.
1. Begin capturing packets between the TOE and the remote TLS server. 2. Run the tool using the corresponding the tool filter for the first subtest in this test on the MITM test system by executing the command. 3. Initiate a connection from the TOE to the server such that the tool modifies the appropriate packet. 4. Stop capturing packets between the TOE and the remote TLS server. 5. Confirm the expected behavior for this subtest occurred. 6. Repeat Steps 2-5 for each of the remaining subtests, using the appropriate the tool filter.	
Test Results:	The evaluator demonstrated that for each modification to the traffic, that the connection was unsuccessful. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

014	**FCS_TLSC_EXT.1.2 – TLS Client Protocol**
Test Objective:	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 1: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails.

	Note that some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1.
1. On the TOE, modify the reference identifier such that it doesn't match the presented certificate CN. 2. Begin capturing packets between the TOE and the remote TLS server. 3. Stimulate TOE to force a connection to the remote TLS server. 4. Stop capturing packets between the TOE and the remote TLS server. 5. Analyze the packet capture to verify that the connection was unsuccessful.	
Test Results:	The evaluator demonstrated that a server certificate lacking the SAN extension whose CN does not match the reference identifier is rejected by the TOE. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

015	**FCS_TLSC_EXT.1.2 – TLS Client Protocol**
Test Objective:	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 2: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type.
1. On the TOE, modify the reference identifier such that it doesn't match the presented SAN identifier, but does match the presented CN identifier. 2. Begin capturing packets between the TOE and the remote TLS server. 3. Stimulate TOE to force a connection to the remote TLS server. 4. Stop capturing packets between the TOE and the remote TLS server. 5. Analyze the packet capture to verify that the connection was unsuccessful.	
Test Results:	The evaluator demonstrated that a server certificate whose CN matches the reference identifier, but whose SAN extension does not match the reference identifier is rejected by the TOE. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

016	**FCS_TLSC_EXT.1.2 – TLS Client Protocol**
Test Objective:	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 3: [conditional] If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.
1. Configure the TOE such that the reference identifier matches the CN. 2. Begin capturing packets between the TOE and the remote TLS server. 3. Stimulate TOE to force a connection to the remote TLS server. 4. Stop capturing packets between the TOE and the remote TLS server. 5. Analyze packet capture to verify that the server presented the certificate created during the Setup and that the connection was successful.	
Test Results:	The evaluator demonstrated that a server certificate whose CN matches the reference identifier and lacks the SAN extension is accepted by the TOE. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

017	**FCS_TLSC_EXT.1.2 – TLS Client Protocol**
Test Objective:	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:

	Test 4: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds.
1. On the TOE, modify the reference identifier such that it doesn't match the presented CN identifier, but does match the presented SAN identifier. 2. Begin capturing packets between the TOE and the remote TLS server. 3. Stimulate TOE to force a connection to the remote TLS server. 4. Stop capturing packets between the TOE and the remote TLS server. 5. Analyze the packet capture to verify that the connection was successful.	
Test Results:	The evaluator demonstrated that a server certificate whose CN does not match the reference identifier, but whose SAN does match the reference identifier is accepted by the TOE. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual
018	**FCS_TLSC_EXT.1.2 – TLS Client Protocol**
Test Objective:	Test 5: The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. Test 5.1: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails. Test 5.2: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.come) and verify that the connection fails. Test 5.3: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails. Test 5.4: [conditional]: If wildcards are not supported, the evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection fails.
Tests 5.1 - 5.3: N/A – According to the Security Target, wildcards are not supported by the TOE. **Test 5.4:** 1. On the server, install a certificate containing a wildcard in the left-most label (e.g. *.catl.local), and specify the reference identifier of the host to be with a single left-most label. 2. Begin capturing packets between the TOE and the server. 3. Connect the TOE to the server. 4. Stop capturing packets between the TOE and the server. 5. Verify the connection fails. 6. Repeat Steps 1-5 for each supported type of reference identifier.	
Test Results:	The evaluator confirmed that wildcards are not supported by the TOE by presenting a server

	certificate containing a wildcard in the left-most label that matched the reference identifier. The connection was rejected by the TOE. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

019	**FCS_TLSC_EXT.1.2 – TLS Client Protocol**
Test Objective:	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 6: [conditional] If URI or Service name reference identifiers are supported, the evaluator shall configure the DNS name and the service identifier. The evaluator shall present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator shall repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails.
N/A – Per the ST, the TOE does not support URI or Service name reference identifiers.	
Test Results:	N/A
Execution Method:	N/A

020	**FCS_TLSC_EXT.1.2 – TLS Client Protocol**
Test Objective:	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 7: [conditional] If pinned certificates are supported the evaluator shall present a certificate that does not match the pinned certificate and verify that the connection fails.
N/A – Per the ST, the TOE does not support pinned certificates.	
Test Results:	N/A
Execution Method:	N/A

021	**FCS_TLSC_EXT.1.3 – TLS Client Protocol**
Test Objective:	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 1: The evaluator shall demonstrate that a server using a certificate without a valid certification path results in an authentication failure. Using the administrative guidance, the evaluator shall then load the trusted CA certificate(s) needed to validate the server's certificate, and demonstrate that the connection succeeds. The evaluator then shall delete one of the CA certificates, and show that the connection fails.
1. Confirm the invalid certificate path (CA missing as Root Trust Anchor) is correctly configured in the server.conf file. 2. Begin capturing packets between the TOE and the server. 3. Connect the TOE to the server. 4. Stop capturing packets between the TOE and the server. 5. Verify the connection fails. 6. Modify the server.conf to use the valid certificate path (correct CA is defined as Root Trust Anchor) . 7. Repeat steps 2-4 and verify the connection succeeds. 8. Configure server to send invalidCertChain (intermediate CA deleted) 9. Repeat steps 2-4 and verify the connection fails.	
Test Results:	The evaluator demonstrated that when the trusted CA certificate(s) needed to validate the server's certificate are present in the TOE's trust store database, the presented server certificate is accepted. When one of the trusted CA certificates needed to validate the server's certificate is not contained within the trust store database, the presented server certificate is not accepted. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

022	**FCS_TLSC_EXT.1.3 – TLS Client Protocol**
Test Objective:	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 2: The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure.
This test assurance activity is met by testing performed in FIA_X509_EXT.1.1 – Test 3 – (Test Case 068).	
Test Results:	This test assurance activity is met by testing performed in FIA_X509_EXT.1.1 – Test 3 – (Test Case 068).
Execution Method:	Manual

023	**FCS_TLSC_EXT.1.3 – TLS Client Protocol**
Test Objective:	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 3: The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure.
This test assurance activity is met by testing performed in FIA_X509_EXT.1.1 – Test 2 (Test Case 067).	
Test Results:	This test assurance activity is met by testing performed in FIA_X509_EXT.1.1 – Test 2 (Test Case 067).
Execution Method:	Manual

024	**FCS_TLSC_EXT.1.3 – TLS Client Protocol**
Test Objective:	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 4: The evaluator shall demonstrate that a server using a certificate which does not have a valid identifier results in an authentication failure.
This test assurance activity is met by testing performed in FCS_TLSC_EXT.1.2 – Test 1, Test 2 (Test Case 014, 015).	
Test Results:	This test assurance activity is met by testing performed in FCS_TLSC_EXT.1.2 – Test 1, Test 2 (Test Case 014, 015).
Execution Method:	Manual

032	**FCS_TLSC_EXT.2.1 – TLS Client Protocol**
Test Objective:	The evaluator shall also perform the following tests: Test 1: The evaluator shall establish a connection to a server that is not configured for mutual authentication (i.e. does not send Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake.
1. Ensure the action in the Setup is performed. 2. Begin capturing packets between the TOE and the remote TLS server. 3. Stimulate TOE to force a connection to the remote TLS server. 4. Stop capturing packets between the TOE and the remote TLS server. 5. Analyze the packet capture to verify that the expected results were satisfied.	
Test Results:	The TOE does not send a client certificate message when the server is not configured for mutual authentication. This was performed for the Splunk Forwarder to Splunk Indexer channel.
Execution Method:	Manual

033	**FCS_TLSC_EXT.2.1 – TLS Client Protocol**
Test Objective:	The evaluator shall also perform the following tests:

	Test 2: The evaluator shall establish a connection to a server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.
1. Ensure the action in the Setup is performed. 2. Begin capturing packets between the TOE and the remote TLS server. 3. Stimulate TOE to force a connection to the remote TLS server. 4. Stop capturing packets between the TOE and the remote TLS server. 5. Analyze the packet capture to verify that the expected results were satisfied.	
Test Results:	The TOE sends a client certificate message when the server is configured for mutual authentication. This was performed for the Splunk Forwarder to Splunk Indexer channel.
Execution Method:	Manual

041	**FCS_TLSC_EXT.5.1 – TLS Client Protocol**
Test Objective:	The evaluator shall perform the following test: Test 1: The evaluator shall configure a server to perform key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.
1. Configure the server to perform an ECDHE key exchange using the secp256r1 curve onto the remote TLS server. 2. Begin capturing packets between the TOE and the remote TLS server. 3. Stimulate TOE to force a connection to the remote TLS server. 4. Stop capturing packets between the TOE and the remote TLS server. 5. Analyze the packet capture to verify that the named curve in the Server Key Exchange message corresponds with the one specified in Step 1 and that the TLS connection was successful. 6. Repeat Steps 1-5, except replace "secp256r1" with "secp384r1". 7. Repeat Steps 1-5, except replace "secp256r1" with "secp521r1".	
Test Results:	The TOE successfully establishes a TLS connection to the server using each of the TOE's supported curves and/or groups: secp256r1 (prime256v1), secp384r1, secp521r1. This was performed for the Splunk to SMTP server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

042	**FCS_TLSS_EXT.1.1 – TLS Server Protocol**
Test Objective:	The evaluator shall also perform the following tests: Test 1: The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Prepare the environment for capturing packets between the management workstation and the TOE (Indexer). 1. Begin capturing packets between the management workstation and the TOE (Indexer). 2. Attempt to establish a TLS connection to the TOE using the following ciphersuite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (ECDHE-ECDSA-AES256-SHA384) 3. Stop capturing packets. 4. Verify the connection succeeded. 5. Repeat Steps 1-4, except in Step 1 use the "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256" (ECDHE-ECDSA-AES128-GCM-SHA256) cipher suite. 6. Repeat Steps 1-4, except in Step 1 use the "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384"	

(ECDHE-ECDSA-AES256-GCM-SHA384) cipher suite.

Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer).

1. Begin capturing packets between the client external Trusted Data Feed (Forwarder) and the TOE (Indexer).
2. Attempt to establish a TLS connection to the TOE using the following ciphersuite:

   TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (ECDHE-ECDSA-AES256-SHA384)

3. Stop capturing packets.
4. Verify the connection succeeded.
5. Repeat Steps 1-4, except in Step 1 use the "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256" (ECDHE-ECDSA-AES128-GCM-SHA256) cipher suite.
6. Repeat Steps 1-4, except in Step 1 use the "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384" (ECDHE-ECDSA-AES256-GCM-SHA384) cipher suite.

Test Results:	The evaluator established a TLS connection using each of the claimed TLS cipher suites. This was performed for the Test Workstation to Splunk Indexer web server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

043	**FCS_TLSS_EXT.1.1 – TLS Server Protocol**
Test Objective:	The evaluator shall also perform the following tests: Test 2: The evaluator shall send a Client Hello to the server with a list of cipher suites that does not contain any of the cipher suites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the server denies the connection.

Three unsupported ciphersuites in Client Hello

Prepare the environment for capturing packets between the management workstation and the TOE (Indexer).

1. Begin capturing packets between the TOE and the client management workstation.
2. From the management workstation, attempt to establish a TLS connection to the TOE using the following ciphersuites in the Client Hello:

   TLS_RSA_WITH_AES_128_CBC_SHA
   TLS_RSA_WITH_AES_128_CBC_SHA256
   TLS_RSA_WITH_AES_256_CBC_SHA

3. Stop capturing packets.
4. Verify the connection attempt failed.

Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer).

1. Begin capturing packets between the TOE and the external Trusted Data Feed (Forwarder).
2. From the external Trusted Data Feed (Forwarder), attempt to establish a TLS connection to the TOE using the following ciphersuites in the Client Hello:

   TLS_RSA_WITH_AES_128_CBC_SHA
   TLS_RSA_WITH_AES_128_CBC_SHA256
   TLS_RSA_WITH_AES_256_CBC_SHA

3. Stop capturing packets.
4. Verify the connection attempt failed.

TLS_NULL_WITH_NULL_NULL only in Client Hello

Prepare the environment for capturing packets between the management workstation and the TOE (Indexer).

1. Begin capturing packets between the TOE and the client management workstation.
2. Execute the command to perform the man-in-the-middle modification (MITM) to the Client Hello such that the "TLS_NULL_WITH_NULL_NULL" ciphersuite is presented.
3. Initiate a TLS connection from the test machine to the TOE by executing the command.
4. Stop capturing packets.
5. Verify that the connection attempt failed.

Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer).

1. Begin capturing packets between the TOE and the client external Trusted Data Feed.
2. Execute the command to perform the man-in-the-middle modification (MITM) to the Client Hello such that the "TLS_NULL_WITH_NULL_NULL" ciphersuite is presented:
3. Initiate a TLS connection from the test machine to the TOE by executing the command.
4. Stop capturing packets.
5. Verify that the connection attempt failed.

Test Results:	The evaluator presented a Client Hello containing a list of disallowed cipher suites to the TOE. The TOE rejected the connection. The evaluator then presented a Client Hello containing the "TLS_NULL_WITH_NULL_NULL" cipher suite to the TOE. The TOE also rejected that connection. This was performed for the Test Workstation to Splunk Indexer web server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

044	**FCS_TLSS_EXT.1.1 – TLS Server Protocol**
Test Objective:	The evaluator shall also perform the following tests: Test 3: If RSA key exchange is used in one of the selected ciphersuites, the evaluator shall use a client to send a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake. The evaluator shall verify that the handshake is not completed successfully and no application data flows.
N/A – RSA key exchange is not used in any of the selected cipher suites.	
Test Results:	N/A
Execution Method:	N/A

045	**FCS_TLSS_EXT.1.1 – TLS Server Protocol**
Test Objective:	The evaluator shall also perform the following tests: Test 4: The evaluator shall perform the following modifications to the traffic: Test 4.1: Change the TLS version proposed by the client in the Client Hello to a non-supported TLS version (for example 1.3 represented by the two bytes 03 04) and verify that the server rejects the connection. Test 4.2: Modify a byte in the data of the client's Finished handshake message, and verify that the server rejects the connection and does not send any application data. Test 4.3: Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption): Generate a Fatal Alert by sending a Finished message from the client before the client sends a ChangeCipherSpec message, and then send a Client Hello with the session identifier from the previous incomplete session, and verify that the server does not resume the session.

	Test 4.4: Send a message consisting of random bytes from the client after the client has issued the ChangeCipherSpec message and verify that the server denies the connection.
Prepare the environment for capturing packets between the management workstation and the TOE (Indexer). 1. Begin capturing packets between the TOE and the client management workstation. 2. Run the tool using the corresponding the tool filter for the first subtest in this test on the MITM test system by executing the command. 3. Initiate a connection from the client to the TOE such that the tool modifies the appropriate packet. 4. Stop capturing packets between the TOE and the remote TLS client. 5. Confirm the expected behavior for this subtest occurred. 6. Repeat Steps 2-5 for each of the remaining subtests, using the appropriate the tool filter. Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 1. Begin capturing packets between the TOE and the client external Trusted Data Feed (Forwarder). 2. Run the tool using the corresponding the tool filter for the first subtest in this test on the MITM test system by executing the command. 3. Initiate a connection from the client to the TOE such that the tool modifies the appropriate packet. 4. Stop capturing packets between the TOE and the remote TLS client. 5. Confirm the expected behavior for this subtest occurred. 6. Repeat Steps 2-5 for each of the remaining subtests, using the appropriate the tool filter.	
Test Results:	The evaluator performed the modification to the traffic for each of the cases described. In each instance, the TLS connection was rejected by the TOE. This was performed for the Test Workstation to Splunk Indexer web server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

046	**FCS_TLSS_EXT.1.2 – TLS Server Protocol**
Test Objective:	Test 1: The evaluator shall send a Client Hello requesting a connection with version SSL 2.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 3.0 and TLS 1.0, and TLS 1.1 if it is selected.
Prepare the environment for capturing packets between the management workstation and the TOE (Indexer). 1. Begin capturing packets between the TOE and client management workstation. 2. Execute the command on the management workstation to initiate a connection to the TOE using the disallowed protocols. 3. Stop capturing packets and verify that the connection(s) failed for the mandatory and selected protocol versions in the SFR. Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 1. Begin capturing packets between the TOE and external Trusted Data Feed (Forwarder). 2. Execute the command on the external Trusted Data Feed (Forwarder) to initiate a connection to the TOE using the disallowed protocols. 3. Stop capturing packets and verify that the connection(s) failed for the mandatory and selected protocol versions in the SFR.	
Test Results:	The evaluator sent a Client Hello requesting a connection using SSL 2.0, SSL 3.0, TLS 1.0, and TLS 1.1. In each of these cases, the connection was denied by the TOE. This was performed for the Test Workstation to Splunk Indexer web server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

047	**FCS_TLSS_EXT.1.3 – TLS Server Protocol**
Test Objective:	The evaluator shall conduct the following tests. The testing can be carried out manually with a packet analyzer or with an automated framework that similarly captures such empirical

	evidence. Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. Test 1: [conditional] If RSA-based key establishment is selected, the evaluator shall configure the TOE with a certificate containing a supported RSA size and attempt a connection. The evaluator shall verify that the size used matches that which is configured and that the connection is successfully established. The evaluator shall repeat this test for each supported size of RSA-based key establishment. Test 2: [conditional] If finite-field (i.e. non-EC) Diffie-Hellman ciphers are selected, the evaluator shall attempt a connection using a Diffie-Hellman key exchange with a supported parameter size or supported group. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported parameter size or group. Test 3: [conditional] If ECDHE ciphers are selected, the evaluator shall attempt a connection using an ECDHE ciphersuite with a supported curve. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported elliptic curve.
Prepare the environment for capturing packets between the management workstation and the TOE (Indexer). 1. Test 1 and Test 2 are N/A due to the condition not being met. 2. Configure the TOE to perform an ECDHE key exchange using the secp256r1 (prime256v1) curve. 3. Begin capturing packets between the management workstation and the TOE. 4. Stimulate the test workstation to force a connection to the TOE. 5. Stop capturing packets between the test client and the TOE. 6. Analyze the packet capture to verify that the named curve in the Server Key Exchange message corresponds with the one specified in Step 1 and that the TLS connection was successful. 7. Repeat Steps 1-5, except replace “secp256r1” with “secp384r1”. 8. Repeat Steps 1-5, except replace “secp256r1” with “secp521r1”. Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 1. Test 1 and Test 2 are N/A due to the condition not being met. 2. Configure the TOE (Indexer) to perform an ECDHE key exchange using the secp256r1 (prime256v1) curve. 3. Begin capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 4. Stimulate the external Trusted Data Feed (Forwarder) to force a connection to the TOE (Indexer). 5. Stop capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 6. Analyze the packet capture to verify that the named curve in the Server Key Exchange message corresponds with the one specified in Step 1 and that the TLS connection was successful. 7. Repeat Steps 1-5, except replace “secp256r1” with “secp384r1”. 8. Repeat Steps 1-5, except replace “secp256r1” with “secp521r1”.	
Test Results:	The evaluator confirmed that conditions for Test 1 and Test 2 were not met. As such, those Tests were not performed. In Test 3, the evaluator established a TLS connection using each of the supported elliptic curves and verified that the key agreement parameters in the Key Exchange message were the ones configured. This was performed for the Test Workstation to Splunk Indexer web server and Splunk Forwarder to Splunk Indexer channels.
Execution Method:	Manual

049	**FCS_TLSS_EXT.2.2 – TLS Server Support for Mutual Authentication**
Test Objective:	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity: Test 1: The evaluator shall configure the server to send a certificate request to the client. The

	client shall send a certificate_list structure which has a length of zero. The evaluator shall verify that the handshake is not finished successfully and no application data flows.
Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 1. Begin capturing packets between the TOE and external Trusted Data Feed (Forwarder). 2. Execute the following commands on the external Trusted Data Feed (Forwarder) to initiate a connection to the TOE. 3. Stop capturing packets and verify that the TLS handshake is not finished successfully and that no application data flows.	
Test Results:	The evaluator confirmed that the TLS handshake was not finished successfully, and that no application data flowed after configuring the server to send a certificate request to the client and ensuring that the client sent a certificate_list structure with a length of zero. This was for the Splunk Forwarder to Splunk Indexer channel.
Execution Method:	Manual

050	**FCS_TLSS_EXT.2.2 – TLS Server Support for Mutual Authentication**
Test Objective:	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity: Test 2: The evaluator shall configure the server to send a certificate request to the client. The client shall send no client certificate message, and instead send a client key exchange message in an attempt to continue the handshake. The evaluator shall verify that the handshake is not finished successfully and no application data flows.
Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 1. Begin capturing packets between the TOE (Indexer) and external Trusted Data Feed (Forwarder) 2. Execute the command to perform the man-in-the-middle modification (MITM) such that the client Certificate message is not sent and instead a client Key Exchange message is transmitted. 3. Execute the following commands on the external Trusted Data Feed (Forwarder) to initiate a connection to the TOE. 4. Stop capturing packets and verify that the TLS handshake is not finished successfully and that no application data flows.	
Test Results:	The evaluator confirmed that the TLS handshake was not finished successfully, and that no application data flowed after configuring the server to not send a client certificate message, but to instead send a client key exchange message. This was performed for the Splunk Forwarder to Splunk Indexer channel.
Execution Method:	Manual

051	**FCS_TLSS_EXT.2.2 – TLS Server Support for Mutual Authentication**
Test Objective:	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity: Test 3: The evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify that the handshake is not finished successfully and no application data flows.

Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 1. Begin capturing packets. 2. Run the tool using the corresponding the tool filter such that the supported_signature_algorithm used by the client certificate is incompatible with the presented client certificate. 3. Initiate a connection from the client to the TOE such that the tool modifies the appropriate packet. 4. Stop capturing packets. 5. Confirm the expected behavior (the TLS handshake is not finished successfully and no application data flows).	
Test Results:	The evaluator confirmed that the TLS handshake was not finished successfully, and that no application data flowed after configuring the server to send a certificate request to the client without the supported_signature_algiorithm used by the client's certificate. This was performed for the Splunk Forwarder to Splunk Indexer channel.
Execution Method:	Manual

052	**FCS_TLSS_EXT.2.2 – TLS Server Support for Mutual Authentication**
Test Objective:	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity: Test 4: The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the administrative guidance, the evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails.
Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 1. Begin capturing packets between the TOE and the client external Trusted Data Feed (Forwarder). 2. From the external Trusted Data Feed (Forwarder), attempt to establish a TLS connection to the TOE. 3. Stop capturing packets. 4. Verify the connection attempt is successful. 5. Remove the intermediate CA certificate issued by the root CA from the file pointed to by "-CAfile". 6. Repeat Steps 1 – 3 and verify the connection attempt is unsuccessful.	
Test Results:	The evaluator demonstrated that using a certificate without a valid certificate path (by removing one of the intermediate CA certificates in the presented certificate chain) resulted in the failure of the TLS connection. When the complete, valid certificate path was presented, the TLS connection was successful. This was performed for the Splunk Forwarder to Splunk Indexer channel.
Execution Method:	Manual

053	**FCS_TLSS_EXT.2.2 – TLS Server Support for Mutual Authentication**
Test Objective:	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity: Test 5: The aim of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To carry out this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognised by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not in fact correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not in fact terminate in the claimed CA certificate). The evaluator shall verify that the attempted connection is denied.

1. Begin capturing packets. 2. Configure the client to present a valid, trusted certificate chain to the TOE. a. Initiate a connection from the client to the TOE. 3. Stop capturing packets. 4. Verify that the connection is successful. 5. Begin capturing packets. 6. Configure the client to present the "imposter CA" signed certificate and chain to the TOE. a. Initiate a connection from the client to the TOE. 7. Verify that the attempted connection is denied (no application data flows).	
Test Results:	The evaluator verified that the connection was successful when a valid, trusted certificate was presented to the TOE. The evaluator then presented a client certificate whose issuer field matched a CA recognized by the TOE as a trusted CA, but where the key used for the signature on the client certificate did not correspond to the CA certificate trusted by the TOE. In this instance, the TOE rejected the presented client certificate and subsequently denied the connection. This was performed for the Splunk Forwarder to Splunk Indexer channel.
Execution Method:	Manual

054	**FCS_TLSS_EXT.2.2 – TLS Server Support for Mutual Authentication**
Test Objective:	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity: Test 6: The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that the server denies the connection. Ideally, the two certificates should be identical except for the Client Authentication purpose.
Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 1. Begin capturing packets between the TOE and the client external Trusted Data Feed (Forwarder). 2. From the external Trusted Data Feed (Forwarder), attempt to establish a TLS connection to the TOE. 3. Stop capturing packets. 4. Verify the connection attempt is successful. 5. Replace the client certificate and its corresponding client key with a client certificate missing the Client Authentication purpose in the extendedKeyUsage field. 6. Repeat Steps 1 – 3 and verify the connection attempt is unsuccessful.	
Test Results:	The evaluator verified that the TLS connection was successful when a presented client certificate contained the Client Authentication purpose. The evaluator then verified that the TLS connection was unsuccessful when a presented client certificate lacked the Client Authentication purpose. This was performed for the Splunk Forwarder to Splunk Indexer channel.
Execution Method:	Manual

055	**FCS_TLSS_EXT.2.2 – TLS Server Support for Mutual Authentication**
Test Objective:	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity: Test 7: The evaluator shall perform the following modifications to the traffic: a) Configure the server to require mutual authentication and then modify a byte in the client's certificate. The evaluator shall verify that the server rejects the connection. b) Configure the server to require mutual authentication and then modify a byte in the signature block of the client's Certificate Verify handshake message. The evaluator shall verify that the server rejects the connection.

Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 1. Begin capturing packets between the TOE and the client external Trusted Data Feed. 2. Run the tool using the corresponding the tool filter for the first subtest in this test on the MITM test system by executing the command. 3. Initiate a connection from the client external Trusted Data Feed to the TOE such that the tool modifies the appropriate packet. 4. Stop capturing packets between the management workstation and the TOE. 5. Confirm the expected behavior for this subtest occurred. 6. Repeat Steps 1-5 for each of the remaining subtests, using the appropriate the tool filter.	
Test Results:	The evaluator verified that for each modification to the traffic, the connection was rejected. This was performed for the Splunk Forwarder to Splunk Indexer channel.
Execution Method:	Manual

056	**FCS_TLSS_EXT.2.3 – TLS Server Support for Mutual Authentication**
Test Objective:	Test 1: The evaluator shall send a client certificate with an identifier that does not match any of the expected identifiers and verify that the server denies the connection. The matching itself might be performed outside the TOE (e.g. when passing the certificate on to a directory server for comparison).
Prepare the environment for capturing packets between the external Trusted Data Feed (Forwarder) and the TOE (Indexer). 1. Begin capturing packets between the TOE and the client external Trusted Data Feed (Forwarder). 2. From the external Trusted Data Feed (Forwarder), attempt to establish a TLS connection to the TOE. 3. Stop capturing packets. 4. Verify the connection attempt is unsuccessful.	
Test Results:	The evaluator verified that the TOE rejected the connection when the presented client certificate's identifier did not match any of the expected identifiers. This was performed for the Splunk Forwarder to Splunk Indexer channel.
Execution Method:	Manual

5.3.2 User Data Protection

061	**FDP_DAR_EXT.1.1 - Encryption Of Sensitive Application Data**
Test Objective:	Evaluation activities (after the identification of the sensitive data) are to be performed on all sensitive data listed that are not covered by FCS_STO_EXT.1. The evaluator shall inventory the filesystem locations where the application may write data. The evaluator shall run the application and attempt to store sensitive data. The evaluator shall then inspect those areas of the filesystem to note where data was stored (if any), and determine whether it has been encrypted. If "leverage platform-provided functionality" is selected, the evaluation activities will be performed as stated in the following requirements, which vary on a per-platform basis. For Linux: The Linux platform currently does not provide data-at-rest encryption services which depend upon invocation by application developers. The evaluator shall verify that the Operational User Guidance makes the need to activate platform encryption clear to the end user.
Run on both TOE Indexer and TOE Forwarder 1. Ensure that the LUKS partition is applied by inventorying the directory structure: As root, issue the command: df –l	

2. Verify that the /opt/splunk and /etc/opt/splunk directories are mapped to a LUKS partition by looking for a filesystem marked with splunkhome and splunketc. 3. Verify each LUKS partition identified from the command executed in Step 1: As root, issue the commands: cryptsetup -v status splunkhome cryptsetup -v status splunketc 4. Verify that each partition is appropriately designated as LUKS. 5. Verify that the Operational User Guidance (AGD) makes the need to activate platform encryption clear to the end user.	
Test Results:	The evaluator verified that platform encryption was configured and that the Operational User Guidance makes the need to activate platform encryption clear to the end user in Section 7.8.
Execution Method:	Manual

062	**FDP_DEC_EXT.1.1 – Access to Platform Resources**
Test Objective:	The evaluator shall perform the platform-specific actions below and inspect user documentation to determine the application's access to hardware resources. The evaluator shall ensure that this is consistent with the selections indicated. The evaluator shall review documentation provided by the application developer and for each resource which it accesses, identify the justification as to why access is required. For Linux: The evaluator shall verify that either the application software or its documentation provides a list of the hardware resources it accesses.
Based on the selection of Linux as the OS there are no testing activities. All activities for Linux are evaluation of the operational guidance and/or the ST.	
Test Results:	Based on the selection of Linux as the OS there are no testing activities. All activities for Linux are evaluation of the operational guidance and/or the ST. Section 5.2, Table 1 of the AGD states that for the Host Platform component, the TOE requires network resources from the host platform.
Execution Method:	N/A

063	**FDP_DEC_EXT.1.2 – Access to Platform Resources**
Test Objective:	The evaluator shall perform the platform-specific actions below and inspect user documentation to determine the application's access to sensitive information repositories. The evaluator shall ensure that this is consistent with the selections indicated. The evaluator shall review documentation provided by the application developer and for each sensitive information repository which it accesses, identify the justification as to why access is required. For Linux: The evaluator shall verify that either the application software or its documentation provides a list of sensitive information repositories it accesses.
N/A – The ST selects "no sensitive information repositories" for this SFR.	
Test Results:	N/A – The ST selects "no sensitive information repositories" for this SFR.
Execution Method:	N/A

064	**FDP_NET_EXT.1.1 – Network Communications**
Test Objective:	The evaluator shall perform the following tests: Test 1: The evaluator shall run the application. While the application is running, the evaluator shall sniff network traffic ignoring all non-application associated traffic and verify that any

	network communications witnessed are documented in the TSS or are user-initiated.

For this test, both the configurations (Indexer and Forwarder) are considered the TOE. This is so all network communications are captured for both of the TOE configurations as the operate within their evaluated configuration as part of the entire defined environment.

1. Begin capturing packets from the TOE (Indexer).
2. Start the TOE (Indexer) application services.
3. Restart the TOE Forwarder operating as a Trusted Data Feed, such that it initiates a connection to the TOE Indexer.
4. Perform the following activities:
   a. Navigate and authenticate to the TOE's Indexer web UI administration URL:

      ```
      https://<TOE-IP-Address>:8000
      ```

   b. Once authenticated: Perform a query that initiates an e-mail alert to the configured SMTP server:

      ```
      index=_internal | head 5 | sendmail to=example@splunk.com
      server=splunk2022-smtp.catl.local:25 subject="Here is an email from
      Splunk" message="This is an example message" use_ssl=true
      use_tls=false
      ```

   c. Execute the following command to list network connections on the TOE Indexer and TOE Forwarder operating as a Trusted Data Feed:

      ```
      ss -tulpna
      ```

      OR

      ```
      netstat -tulpna
      ```

5. Stop capturing packets from entire testbed (two TOE instances).
   Inspect the packet capture and verify that any application associated network communications witnessed are documented in the TSS or are user-initiated.

Test Results:	The evaluator performed this test by first capturing packets, then running the application in both configurations. While the application was running, the evaluator performed activities to stimulate network connections that the application either initiated or responded. The evaluator confirmed that any network communication witnessed was either documented in the TSS or was user-initiated.
Execution Method:	Manual

065	**FDP_NET_EXT.1.1 – Network Communications**
Test Objective:	The evaluator shall perform the following tests: Test 2: The evaluator shall run the application. After the application initializes, the evaluator shall run network port scans to verify that any ports opened by the application have been captured in the ST for the third selection and its assignment. This includes connection-based protocols (e.g. TCP, DCCP) as well as connectionless protocols (e.g. UDP).

TOE configured as Indexer:

1. Start the TOE Indexer application services.
2. After the application is initialized, execute the following command to initiate a network-based port scan against the TOE.
3. Verify that any ports opened by the TOE have been captured in the ST for the third selection and its assignment for this test assurance activity SFR.

TOE configured as Forwarder:

1. Start the TOE Forwarder application services.

2. After the application is initialized, execute the following command to initiate a network-based port scan against the TOE. a. Verify that any ports opened by the TOE have been captured in the ST for the third selection and its assignment for this test assurance activity SFR.	
Test Results:	The evaluator performed this test by starting the TOE application. After the application initialized, the evaluator ran a network-based port scan. The evaluator verified that any ports opened by the application were captured in the ST for the third selection and its assignment.
Execution Method:	Manual

5.3.3 Identification and Authentication

066	**FIA_X509_EXT.1.1 – X.509 Certificate Validation**
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 1: The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn: • by establishing a certificate path in which one of the issuing certificates is not a CA certificate, • by omitting the basicConstraints field in one of the issuing certificates, • by setting the basicConstraints field in an issuing certificate to have CA=False, • by omitting the CA signing bit of the key usage field in an issuing certificate, and • by setting the path length field of a valid CA field to a value strictly less than the certificate path. The evaluator shall then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates, and show that the function fails.

Valid / invalid certificate path

1. Load the Root CA that validates the remote peer's node certificate into the TOE's trusted CA database.
2. Begin capturing packets between the TOE and the remote peer.
3. Initiate a connection between the TOE and the remote peer.
4. Stop capturing packets between the TOE and the remote peer.
5. Verify that the TOE successfully authenticated to the remote peer and that the certificate was successfully validated.
6. Remove from the TOE's trusted CA database the Root CA certificate that was installed during Step 1.
7. Repeat Steps 2-4.
8. Verify that the TOE failed to successfully authenticate to the remote peer and that the certificate failed to validate.

Not a CA certificate

Refer to FIA_X509_EXT.1.2 – Test 1 and 2 (Test Case 075, 076)

Omitting basicConstraints

Refer to FIA_X509_EXT.1.2 – Test 1 (Test Case 075)

Setting CA=FALSE in basicConstraints

Refer to FIA_X509_EXT.1.2 – Test 2 (Test Case 076)

Omitting the CA signing bit

1. Load the Root CA that validates the remote peer's node certificate signed by a CA without the CA signing bit into the TOE's trusted CA database.
2. Begin capturing packets between the TOE and the remote peer.
3. Initiate a connection between the TOE and the remote peer.
4. Stop capturing packets between the TOE and the remote peer.
5. Verify that the TOE failed to successfully authenticate to the remote peer and that the certificate failed to validate.

CA path length value strictly less than certificate path

1. Load the Root CA that validates the remote peer's node certificate signed by an intermediate CA whose issuer intermediate CA path length value is strictly less than the certificate path into the TOE's trusted CA database.
2. Begin capturing packets between the TOE and the remote peer.
3. Initiate a connection between the TOE and the remote peer.
4. Stop capturing packets between the TOE and the remote peer.
5. Verify that the TOE failed to successfully authenticate to the remote peer and that the certificate failed to validate.

Test Results:	The evaluator confirmed that a valid certificate path consisting of valid CA certificates resulted in the successful establishment of the TLS connection. For each of the other scenarios described in the test objective, the evaluator verified that the TLS connection was unsuccessful. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

067	**FIA_X509_EXT.1.1 – X.509 Certificate Validation**
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.
1. Configure the remote peer to present an expired certificate to the TOE. 2. Begin capturing packets between the TOE and the remote peer. 3. Initiate a connection between the TOE and the remote peer. 4. Stop capturing packets between the TOE and the remote peer. 5. Verify that the TOE failed to successfully authenticate to the remote peer and that the certificate failed to validate.	
Test Results:	The evaluator confirmed that the TOE rejected a peer certificate that was deemed expired. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

068	**FIA_X509_EXT.1.1 – X.509 Certificate Validation**
Test Objective:	The tests described must be performed in conjunction with the other certificate services

	evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-"conditional on whether CRL, OCSP, OCSP Stapling or OCSP Multi-stapling is selected; if multiple methods are selected, then the following tests shall be performed for each method: • The evaluator shall test revocation of the node certificate. • The evaluator shall also test revocation of an intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA), if intermediate CA certificates are supported. If OCSP stapling per RFC 6066 is the only supported revocation method, this test is omitted. • The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.
1. Begin capturing packets between the TOE and the TOE and the remote peer. 2. Initiate a connection between the TOE and the remote peer. 3. Stop capturing packets between the TOE and the remote peer. 4. Verify that the TOE successfully authenticated to the remote peer and that the certificate was successfully validated. 5. Revoke the node certificate. 6. Repeat Steps 2-3. 7. Verify that the TOE failed to successfully authenticate to the remote peer and that the certificate failed to validate. 8. Un-revoke the node certificate and then revoke the Intermediate01 CA certificate. 9. Repeat Steps 2-3 and 7.	
Test Results:	The evaluator confirmed that an unrevoked certificate was successfully validated by the TOE. The evaluator then confirmed that a revoked node / leaf certificate was rejected by the TOE and the connection was unsuccessful. Finally, the evaluator confirmed that a revoked intermediate CA certificate was rejected by the TOE and the connection was unsuccessful. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

069	**FIA_X509_EXT.1.1 – TD0669 - X.509 Certificate Validation – TD0669**
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 4: If any OCSP option is selected, the evaluator shall ensure the TSF has no other source of revocation information available and configure the OCSP server or use a man-in-the-middle tool to present an OCSP response signed by a certificate that does not have the OCSP signing purpose and which is the only source of revocation status information advertised by the CA issuing the certificate being validated. The evaluator shall verify that validation of the OCSP response fails and that the TOE treats the certificate being checked as invalid and rejects the connection. If CRL is selected, the evaluator shall likewise configure the CA to be the only source of revocation status information, and sign a CRL with a certificate that does not have the

	cRLsign key usage bit set. The evaluator shall verify that validation of the CRL fails and that the TOE treats the certificate being checked as invalid and rejects the connection. Note: The intent of this test is to ensure a TSF does not trust invalid revocation status information. A TSF receiving invalid revocation status information from the only advertised certificate status provider should treat the certificate whose status is being checked as invalid. This should generally be treated differently from the case where the TSF is not able to establish a connection to check revocation status information, but it is acceptable that the TSF ignore any invalid information and attempt to find another source of revocation status (another advertised provider, a locally configured provider, or cached information) and treat this situation as not having a connection to a valid certificate status provider.
CRL 1. Place a CRL signed by a CA without the cRLsign key usage bit set with no certificates revoked at the CRL distribution point. 2. Initiate a connection between the TOE and the remote peer. 3. Verify the connection to the remote peer is unsuccessful.	
Test Results:	The evaluator confirmed that a CRL signed by a CA without the cRLsign key usage bit set was not validated successfully and that the connection between the TOE and the remote peer was rejected. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

070	**FIA_X509_EXT.1.1 – X.509 Certificate Validation**
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)
1. If not already present, load the Root CA into the TOE's trusted CA database. 2. Begin capturing packets between the TOE and the remote peer. 3. Run the tool using the script to modify the certificate on the MITM test system by executing the command. 4. Initiate a connection between the TOE and the remote peer such that the MITM modification occurs. 5. Stop capturing packets between the TOE and the remote peer. 6. Verify that the TOE fails to validate the certificate.	
Test Results:	The evaluator performed the modification to the certificate and confirmed that the TOE rejected it, and in turn, the TLS connection. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

071	**FIA_X509_EXT.1.1 – X.509 Certificate Validation**
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate

	that the certificate fails to validate. (The signature on the certificate will not validate.)
1. If not already present, load the Root CA into the TOE's trusted CA database. 2. Begin capturing packets between the TOE and the remote peer. 3. Run the tool using the script to modify the certificate on the MITM test system by executing the command. 4. Initiate a connection between the TOE and the remote peer such that the MITM modification occurs. 5. Stop capturing packets between the TOE and the remote peer. 6. Verify that the TOE fails to validate the certificate.	
Test Results:	The evaluator performed the modification to the certificate and confirmed that the TOE rejected it, and in turn, the TLS connection. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

072	**FIA_X509_EXT.1.1 – X.509 Certificate Validation**
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
1. If not already present, load the Root CA into the TOE's trusted CA database. 2. Begin capturing packets between the TOE and the remote peer. 3. Run the tool using the script to modify the certificate on the MITM test system by executing the command. 4. Initiate a connection between the TOE and the remote peer such that the MITM modification occurs. 5. Stop capturing packets between the TOE and the remote peer. 6. Verify that the TOE fails to validate the certificate.	
Test Results:	The evaluator performed the modification to the certificate and confirmed that the TOE rejected it, and in turn, the TLS connection. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

073	**FIA_X509_EXT.1.1 – X.509 Certificate Validation**
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/Sig). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.
1. Create an EC leaf certificate ("leaf"), two EC intermediate CA certificates ("int CA 02" and "int CA 01"), and an EC root CA certificate ("root CA"), such that they are all chained up to the EC root CA certificate: leaf → int CA 02 → int CA 01 → root CA.	

2. Install the "root CA" certificate created in Step 1 into the TOE's trust store such that it is designated as a trust anchor. 3. Load the "leaf", "int CA 02", and "int CA 01" onto the remote endpoint such that they are presented to the TOE when a connection is established between the remote endpoint and the TOE. 4. Initiate a connection between the TOE and the remote endpoint. 5. Verify that the TOE validates the certificate chain (i.e. the connection is successful).	
Test Results:	The evaluator constructed a certificate chain consisting of an EC leaf certificate, an EC intermediate CA certificate, another EC intermediate CA certificate, and an EC root certificate, where the elliptic curves were specified as named curves. The evaluator installed the EC root CA certificate into the TOE's trust store, presented the certificate chain to the TOE, and verified that the TOE validated it successfully, and that the TLS connection was successful. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

074	**FIA_X509_EXT.1.1 – X.509 Certificate Validation**
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 9: (Conditional on support for EC certificates as indicated in FCS_COP.1/Sig). The evaluator shall replace the intermediate certificate in the certificate chain for Test 8a with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 8a, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.
(continuation from FIA_X509_EXT.1.1 – Test 8) 1. Regenerate "int CA 01" with a modified public key information where the EC parameters use an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate, hereafter referred to as: "int CA 01 explicit". Ensure that "int CA 01 explicit" is signed by "root CA" that was created in Step 1, with no other changes. Generate a new leaf certificate: (leaf → int CA 02 → int CA 01 explicit → root CA) a. Execute the command to generate the explicit parameter version of the key generated from using a named curve. 2. Load the "leaf → int CA 02 → int CA 01 explicit" chain onto the remote endpoint such that it is presented to the TOE when a connection is established between the remote endpoint and the TOE. 3. Initiate a connection between the TOE and the remote endpoint. 4. Verify that the TOE treats the certificate chain as invalid (i.e. the connection is unsuccessful).	
Test Results:	The evaluator regenerated the certificate chain consisting of an EC leaf certificate, an EC intermediate CA certificate, another EC intermediate CA certificate, and an EC root certificate, where the elliptic curves were specified as named curves, except that the intermediate EC certificate issued by the EC root certificate was generated with elliptic curves specified using the explicit format version. With the same EC root CA certificate still present in the TOE's trust store, the evaluator presented this new certificate chain to the TOE, and verified that the TOE failed to validate it successfully, and that the TLS connection was unsuccessful. This was

	performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

075	**FIA_X509_EXT.1.2 – X.509 Certificate Validation**
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 1: The evaluator shall ensure that the certificate of at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator shall confirm that validation of the certificate path fails (i) as part of the validation of the peer certificate belonging to this chain; and/or (ii) when attempting to add the CA certificate without the basicConstraints extension to the TOE's trust store.
1. If not already present, load the Root CA into the TOE's trusted CA database. 2. Begin capturing packets between the TOE and the remote peer. 3. Initiate a connection between the TOE and the remote peer. 4. Stop capturing packets between the TOE and the remote peer. 5. Verify that the TOE failed to successfully authenticate to the remote peer and that the certificate failed to successfully validate.	
Test Results:	The evaluator verified that a CA certificate lacking the basicConstraints extension was rejected by the TOE, and that the TLS connection was unsuccessful. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

076	**FIA_X509_EXT.1.2 – X.509 Certificate Validation**
Test Objective:	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 2: The evaluator shall ensure that the certificate of at least one of the CAs in the chain has the CA flag in the basicConstraints extension not set (or set to FALSE). The evaluator shall confirm that validation of the certificate path fails (i) as part of the validation of the peer certificate belonging to this chain; and/or (ii) when attempting to add the CA certificate with the CA flag not set (or set to FALSE) in the basicConstraints extension to the TOE's trust store.
1. If not already present, load the Root CA into the TOE's trusted CA database. 2. Begin capturing packets between the TOE and the remote peer. 3. Initiate a connection between the TOE and the remote peer. 4. Stop capturing packets between the TOE and the remote peer. 5. Verify that the TOE failed to successfully authenticate to the remote peer and that the certificate failed to successfully validate.	
Test Results:	The evaluator verified that a CA certificate that has the CA flag in the basicConstraints extension not set (or set to FALSE) was rejected by the TOE, and that the TLS connection was unsuccessful. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

077	**FIA_X509_EXT.2.2 – X.509 Certificate Validation – Test 1**
Test Objective:	The evaluator shall perform the following test for each trusted channel: Test 1: The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.
1. Ensure the CRL files are present in the non-TOE entity location: /etc/opt/splunk/auth/crl/ 2. Initiate a connection between the TOE and the remote peer. 3. Verify that the TOE successfully authenticated to the remote peer and that the certificate was successfully validated. 4. Manipulate the environment so that the TOE is unable to retrieve the CRLs from the CRL distribution point: a. Execute the following command to clear the local CRL cache from the non-TOE entity: rm /etc/opt/splunk/auth/crl/*.pem 5. Initiate a connection between the TOE and the remote peer. 6. Verify that the TOE successfully authenticated to the remote peer and that the certificate validated.	
Test Results:	The evaluator verified that when the CRL distribution point was accessible (with valid CRL files) by the TOE as part of a TLS connection request, the TOE successfully authenticated to the remote peer and the certificate was successfully validated. The evaluator then manipulated the environment so that the TOE was unable to retrieve the CRL files from the CRL distribution point, initiated a TLS connection, and verified that the TOE successful authenticated to the remote peer and that the certificate was successfully validated. This was performed for the following trusted channels: Splunk Indexer to SMTP server, Splunk Forwarder to Splunk Indexer. In the case of the Splunk Forwarder to Splunk Indexer channel, the client and server certificate were both assessed.
Execution Method:	Manual

078	**FIA_X509_EXT.2.2 – X.509 Certificate Validation**
Test Objective:	The evaluator shall perform the following test for each trusted channel: Test 2: The evaluator shall demonstrate that an invalid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity cannot be accepted.
Invalid certificate testing is performed in FIA_X509_EXT.1 – Test Case 068, 069, and FIA_X509_EXT.2 – Test Case 077.	
Test Results:	Invalid certificate testing is performed in FIA_X509_EXT.1 – Test Case 068 and 069.
Execution Method:	Manual

5.3.4 Security Management

079	**FMT_CFG_EXT.1.1 – Secure by Default Configuration**
Test Objective:	If the application uses any default credentials the evaluator shall run the following tests.

	Test 1: The evaluator shall install and run the application without generating or loading new credentials and verify that only the minimal application functionality required to set new credentials is available.
N/A – There are no default credentials as per the ST, the initial installation of the TOE prompts the security administrator to create a username and password.	
Test Results:	N/A – There are no default credentials as per the ST, the initial installation of the TOE prompts the security administrator to create a username and password.
Execution Method:	Manual

080	**FMT_CFG_EXT.1.1 – Secure by Default Configuration**
Test Objective:	If the application uses any default credentials the evaluator shall run the following tests. Test 2: The evaluator shall attempt to clear all credentials and verify that only the minimal application functionality required to set new credentials is available.
N/A – There are no default credentials as per the ST, the initial installation of the TOE prompts the security administrator to create a username and password.	
Test Results:	N/A – There are no default credentials as per the ST, the initial installation of the TOE prompts the security administrator to create a username and password.
Execution Method:	Manual

081	**FMT_CFG_EXT.1.1 – Secure by Default Configuration**
Test Objective:	If the application uses any default credentials the evaluator shall run the following tests. Test 3: The evaluator shall run the application, establish new credentials and verify that the original default credentials no longer provide access to the application.
N/A – There are no default credentials as per the ST, the initial installation of the TOE prompts the security administrator to create a username and password.	
Test Results:	N/A – There are no default credentials as per the ST, the initial installation of the TOE prompts the security administrator to create a username and password.
Execution Method:	Manual

082	**FMT_CFG_EXT.1.2 – Secure by Default Configuration**
Test Objective:	The evaluator shall install and run the application. The evaluator shall inspect the filesystem of the platform (to the extent possible) for any files created by the application and ensure that their permissions are adequate to protect them. The method of doing so varies per platform. For Linux: The evaluator shall run the command find -L . -perm /002 inside the application's data directories to ensure that all files are not world-writable. The command should not print any files.

For this test, both the configurations (Indexer and Forwarder) are considered the TOE. This is so all TOE filesystem permissions are captured regardless of the TOE configurations (Indexer and Forwarder) as they operate within their evaluated configuration as part of the entire defined environment.

1. Install the Splunk application to the systems.
2. Start the TOE (Indexer) application services and
   Start the TOE (Forwarder) application services to stimulate the communication channel.
3. Perform some activity within the Splunk application:

   - Perform a query on the TOE to cause it to send an e-mail alert to the configured SMTP server:

     index=_internal | head 5 | sendemail to=example@splunk.com server=splunk2022-smtp.catl.local subject="Here is an email from Splunk" message="This is an example message" use_ssl=true use_tls=false

Note: the Trusted Data Feed (Forwarder) when turned on initiates a connection to the TOE exercising functionality. There is no incoming server feeds to the Trusted Data Feed (Forwarder).

4. As root, execute the "find . -perm /002" command inside the Splunk data directories on both the TOE (Indexer) and TOE Forwarder operating as a Trusted Data Feed:

   cd /etc/opt/splunk
   find . -perm /002

   cd /opt/splunk/var/log
   find . -perm /002

   cd /opt/splunk/var/lib
   find . -perm /002

5. Verify that no results were returned after executing the command from Step 4.

Test Results:	The evaluator performed this test by installing and running the Splunk application. After performing some activity within the Splunk application, the evaluator verified that all files present inside the TOE application's data directories were not world-writable.
Execution Method:	Manual

083	**FMT_MEC_EXT.1 – Supported Configuration Mechanism**
Test Objective:	If "invoke the mechanisms recommended by the platform vendor for storing and setting configuration options" is chosen, the method of testing varies per platform as follows: For Linux: The evaluator shall run the application while monitoring it with the utility strace. The evaluator shall make security-related changes to its configuration. The evaluator shall verify that strace logs corresponding changes to configuration files that reside in /etc (for system-specific configuration), in the user's home directory (for user-specific configuration), or /var/lib/ (for configurations controlled by UI and not intended to be directly modified by an administrator).

Indexer:

1. Terminate the TOE Indexer application services.
2. Run the strace utility such that it monitors the TOE Indexer application by executing the following commands:

   runcon -u system_u -t splunk_t -r system_r strace -o strace_splunk -ff /opt/splunk/bin/splunk start

3. Authenticate to the TOE Indexer application via the web GUI.
4. Under the Settings tab select SYSTEM → Server settings → General settings.
5. Make security related changes to the Splunk configuration:
   i. Change name of server to splunk2022-modified-82idx.catl.local.
6. Under the Settings tab select SYSTEM→ Server settings → Email settings.
7. Make security related changes to the Splunk configuration:
   i. Change IP address of email server to 192.168.1.199:25.
8. Terminate the strace utility.
9. Inspect the strace logs and verify that it corresponds to the changes made to files in /etc/opt/splunk.

Forwarder:

1. Terminate the TOE Forwarder application services.
2. Run the strace utility such that it monitors the TOE Forwarder application by executing the following commands:

   runcon -u system_u -t splunk_t -r system_r strace -o strace_splunk -ff /opt/splunk/bin/splunk start

3. Make security related changes to the Splunk configuration: i. Change name of server to splunk2022-modified-82fwd.catl.local. runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk set servername splunk2022-modified-82fwd.catl.local 4. Terminate the strace utility. 5. Inspect the strace logs and verify that it corresponds to the changes made to files in /etc/opt/splunk.	
Test Results:	The evaluator performed this test by running the application while monitoring it with strace. While this was occurring, the evaluator made security-related configuration changes to the application. The evaluator confirmed that strace logs contained output that corresponded to the changes made to relevant configuration files.
Execution Method:	Manual

084	**FMT_SMF.1.1 – Specification of Management Functions**
Test Objective:	The evaluator shall test the application's ability to provide the management functions by configuring the application and testing each option selected from above. The evaluator is expected to test these functions in all the ways in which the ST and guidance documentation state the configuration can be managed.
Enable/disable supported TLS ciphersuites (show both Indexer and Forwarder configurations) 1. Via the TOE platform, specify the "cipherSuite" parameter in the following files to "ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384": /etc/opt/splunk/system/local/web.conf /etc/opt/splunk/system/local/server.conf (Indexer Only): /etc/opt/splunk/system/local/alert_actions.conf /etc/opt/splunk/system/local/inputs.conf (Forwarder Only): /etc/opt/splunk/system/local/outputs.conf 2. Verification of the enabled/disabled ciphersuites is performed during the execution of FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1. **Query the version of both the TOE on both machines** 1. (Forwarder): Execute the following command as the splunk user: runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk version 2. (Indexer): a. Navigate to the TOE web GUI. b. Authenticate to the TOE via the web GUI. c. Navigate to "Help" > "About". d. Verify the TOE version information is displayed.	
Test Results:	The evaluator performed this test by demonstrating that the claimed functions were configurable and/or tested in the ways in which the ST and guidance documentation stated how each configuration is managed.
Execution Method:	Manual

5.3.5 Privacy

085	**FPR_ANO_EXT.1.1 – User Consent for Transmission of Personally Identifiable Information**
Test Objective:	If require user approval before executing is selected, the evaluator shall run the application and exercise the functionality responsibly for transmitting PII and verify that user approval is required before transmission of the PII.

N/A – The ST states that PII is not transmitted over a network.	
Test Results:	N/A – The ST states that PII is not transmitted over a network.
Execution Method:	Manual

5.3.6 Protection of the TSF

086	**FPT_AEX_EXT.1.1 – Anti-Exploitation Capabilities**
Test Objective:	The evaluator shall perform either a static or dynamic analysis to determine that no memory mappings are placed at an explicit and consistent address. The method of doing so varies per platform. For those platforms requiring the same application running on two different systems, the evaluator may alternatively use the same device. After collecting the first instance of mappings, the evaluator must uninstall the application, reboot the device, and reinstall the application to collect the second instance of mappings. For Linux: The evaluator shall run the same application on two different Linux systems. The evaluator shall then compare their memory maps using pmap -x PID to ensure the two different instances share no mapping locations.
1. Install the TOE application on the RedHat Linux system. 2. Install the TOE application on the second RedHat Linux system. 3. Start the Splunk application services on both RedHat Linux systems. 4. Execute the following command, where [PID] is the process ID for the Splunk application on the system, as the root user on both RedHat Linux systems: pmap -x [PID] 5. Examine the memory maps from both systems and ensure that there is no sharing of the same memory locations.	
Test Results:	The evaluator performed this test by installing the TOE application, starting it, then collecting the memory maps. The evaluator then installed the TOE application on a second system, started it, and then collected those memory maps. The evaluator compared the two sets of memory maps and ensured that there was no sharing of the same memory location.
Execution Method:	Manual

087	**FPT_AEX_EXT.1.2 – Anti-Exploitation Capabilities**
Test Objective:	The evaluator shall verify that no memory mapping requests are made with write and execute permissions. The method of doing so varies per platform. The evaluator shall perform static analysis on the application to verify that both • mmap is never be invoked with both the PROT_WRITE and PROT_EXEC permissions, and • mprotect is never invoked with the PROT_EXEC permission.
MMAP 1. In the source code root directory execute the command. grep -R "mmap(" ./ > mmap_calls.txt 2. Examine the output of the command and verify that no memory mapping requests are made using both the PROT_WRITE and PROT_EXEC permissions (barring exceptions claimed in the ST). **MPROTECT** 1. In the source code root directory execute the command. grep -R "mprotect(" ./ > mprotect_calls.txt 2. Examine the output of the command (mprotect_calls.txt) and verify that no memory mapping requests are made	

using the PROT_EXEC permission (barring exceptions claimed in the ST).	
Test Results:	The evaluator examined the TOE application source code and confirmed that no memory mapping requests were made with write and execution permissions.
Execution Method:	Manual

088	**FPT_AEX_EXT.1.3 – Anti-Exploitation Capabilities**
Test Objective:	The evaluator shall configure the platform in the ascribed manner and carry out one of the prescribed tests: For Linux: The evaluator shall ensure that the application can successfully run on a system with either SELinux or AppArmor enabled and in enforce mode.
1. Ensure that SE Linux is enabled and enforcing by executing the sestatus command. 2. Start the TOE application services and conduct the testing throughout the course of this evaluation.	
Test Results:	The evaluator confirmed that SELinux was enabled and enforcing by executing the "sestatus" command and verifying the output.
Execution Method:	Manual

089	**FPT_AEX_EXT.1.4 – Anti-Exploitation Capabilities**
Test Objective:	The evaluator shall run the application and determine where it writes its files. For files where the user does not choose the destination, the evaluator shall check whether the destination directory contains executable files. This varies per platform: For Linux: The evaluator shall run the program, mimicking normal usage, and note where all user-modifiable files are written. The evaluator shall ensure that there are no executable files stored in the same directories to which the application wrote user-modifiable files.
1. Run the Splunk application, mimicking normal usage, and perform several other testing activities throughout the evaluation. 2. As root, execute the following command inside the Splunk installation directories to list all Splunk application executable file paths. 3. Examine the output from Step 2 to determine the list of directories where Splunk application executable files are stored. 4. For each directory determined from Step 3, execute the following command to obtain the last modified date of that directory and all directories recursively within. 5. Execute the following command for each directory where the date is greater than the most recent installation date (RHEL 8.2 instance: Nov 2 2022 13:33; RHEL 7.9 instance: Nov 29 11:44) of the Splunk application. 6. Examine the output from Step 5 and for any directories where no output was returned from the command execution, no executable files are present in them and may be excluded from analysis. 7. For each directory determined in Step 6 to NOT be excluded from further analysis, execute the command. 8. Examine the output from Step 7 and verify that any file output with a date greater than the most recent Splunk application installation date (RHEL 8.2 instance: Nov 2 2022 13:33; RHEL 7.9 instance: Nov 29 11:44) does not have the write permission file attribute.	
Test Results:	The evaluator performed this test by running the application, mimicking normal usage, and noted where all user-modifiable files were written. The evaluator systematically confirmed that no executable files were stored in the same directories to which the application wrote user-modifiable files.
Execution Method:	Manual

090	**FPT_AEX_EXT.1.5 – Anti-Exploitation Capabilities**
Test Objective:	The evaluator will inspect every native executable included in the TOE to ensure that stack-based buffer overflow protection is present. For ELF executables, the evaluator will ensure that each contains references to the symbol **__stack_chk_fail.**
1. Using the source code and a compiler, compile the TOE.	

2. Examine compiler execution output and verify that the TOE was compiled using flags equivalent to GCC's fstack-protector-strong or fstack-protector-all.

Using https://github.com/commoncriteria/canary-detector

1. Survey the installation for executable files (Refer to FPT_TUD_EXT.1.3 – Test 1 (Test Case 096)).
    a. Refer to Step 2 and use TOE_BeforeStart.sha256.list.txt
2. For each native ELF executable included in the TOE, execute:

   python cande.py <executable>

3. Confirm that there is/are references to the symbol: __stack_chk_fail

Test Results:	The evaluator was able to confirm through the analysis of the scanner results and the build_log that the TOE was compiled with using fstack-protector-strong.
Execution Method:	Manual

091	**FPT_IDV_EXT.1 Software Identification and Versions**
Test Objective:	The evaluator shall install the application, then check for the existence of version information. If SWID tags is selected the evaluator shall check for a .swidtag file. The evaluator shall open the file and verify that is contains at least a SoftwareIdentity element and an Entity element.

1. Execute the following command to open and verify that the SWID tag file contains at least a SoftwareIdentity and Entity element:

   cat /opt/splunk/swidtag/splunk-Splunk-Enterprise-primary.swidtag

Test Results:	The evaluator performed this test by installing the application, checking for the existence of version information in a .swidtag file, opened it, and verified that it contained a SoftwareIdentity and Entity element.
Execution Method:	Manual

092	**FPT_API_EXT.1.1 – Use of Supported Services and APIs**
Test Objective:	The evaluator shall then compare the list with the supported APIs (available through e.g. developer accounts, platform developer groups) and ensure that all APIs listed in the TSS are supported.

1. Compare the system call list from the ST and map the system calls to the Unix library (.so).
2. Map the Unix library to the Unix package that the library is contained within (perform Internet searches, lookups).
3. Verify that the Unix libraries and packages are installed on the TOE platform.

Test Results:	The evaluator verified that the list of APIs in the TSS are supported.
Execution Method:	Manual

093	**FPT_LIB_EXT.1 – Use of Third Party Libraries**
Test Objective:	The evaluator shall install the application and survey its installation directory for dynamic libraries. The evaluator shall verify that libraries found to be packaged with or employed by the application are limited to those in the assignment.

1. Evaluator is to create a spreadsheet called FPT_LIB_EXT.1 - library listing verification with the following tabs:
    i. Output of LDD command tab:
    ii. Validate library legitimacy
    iii. Dir Listing search results
    iv. ST vs Found Dyn Lib List Check.

For Tab 1: Investigate dynamic libraries associated with splunkd by issuing the command.

- ldd /opt/splunk/bin/splunkd (output in Column A)
- Using an internet search determine what library package .so is part of. (Identify in Column B)

For Tab 2: Validate the legitimacy of the packages. Make sure that the library isn't identified as malicious code.

- Using internet search determine what the library package is for (Column B)
- Determine if associated with malicious code (Column C)
- Determine if package makes sense to have in product (Column D)

For Tab 3: Ensure all .so were discovered using the LDD command

- Create a recursive directory listing of TOE directories (/etc/opt/splunk and /opt/splunk):

find /etc/opt/splunk/ -type l -regex .*so[09]* && find /opt/splunk/ -type l -regex .*so[09]* && find /etc/opt/splunk/ -type f -regex .*so[09]* && find /opt/splunk/ -type f -regex .*so[09]*

- Search both files for .so files per AA requirement.
- Make a consolidated list of all .so files in column B and then alphabetize
- Determine what library package .so belongs to
- Using the list of libraries packages from Tab 1 color code the libraries of those that were already discovered.
- Check to see if there are any that LDD missed
    i. If yes add library to Tab 2 and verify legitimacy

For Tab 4: Verify that the full list of .so files discovered in Tab 1 and Tab 3 are identified in the ST

- Copy list from ST into column A and alphabetize
- Copy list from Tab 3 and Tab 1 into Column B. Alphabetize and remove any duplicates.
- Compare Column A to Column B and ensure there is a match for all libraries discovered.

Test Results:	The evaluator verified that the list of libraries found on the TOE are declared in the TSS.
Execution Method:	Manual

094	**FPT_TUD_EXT.1.1 – Integrity for Installation and Update**
Test Objective:	The evaluator shall check for an update using procedures described in either the application documentation or the platform documentation and verify that the application does not issue an error. If it is updated or if it reports that no update is available this requirement is considered to be met.
1. Navigate to the TOE web GUI: https://<TOE-IP-Address>:8000 2. Authenticate via the TOE web GUI. 3. Click on "Messages" on the top right-hand menu. 4. Verify that the TOE reports that an update is available. 5. Navigate to the link "Click here for details." 6. Verify that the URL resolves and that the application does not issue an error.	
Test Results:	The evaluator performed this test by checking for a software update and confirming that the TOE reported that an update was available.
Execution Method:	Manual

095	**FPT_TUD_EXT.1.2 – Integrity for Installation and Update**
Test Objective:	The evaluator shall query the application for the current version of the software according to the operational user guidance. The evaluator shall then verify that the current version matches that of the documented and installed version.
1. Query the TOE's current version as the splunk user: runcon -u system_u -t splunk_t -r system_r /opt/splunk/bin/splunk version 2. Verify the output corresponds to that of the documented and installed version.	

Test Results:	The evaluator performed this test by querying the application for the current version of its software. The evaluator confirmed that this version matched that of the documented and installed version.
Execution Method:	Manual

096	**FPT_TUD_EXT.1.3 – Integrity for Installation and Update**
Test Objective:	The evaluator shall verify that the application's executable files are not changed by the application. For all other platforms, the evaluator shall perform the following test: Test 1: The evaluator shall install the application and then locate all of its executable files. The evaluator shall then, for each file, save off either a hash of the file or a copy of the file itself. The evaluator shall then run the application and exercise all features of the application as described in the ST. The evaluator shall then compare each executable file with the either the saved hash or the saved copy of the files. The evaluator shall verify that these are identical.

1. Install the Splunk application to the system:

   Refer to FPT_TUD_EXT.2.2 – (Test Case 100)
   Perform Steps 1 through 8.

2. Calculate and record the SHA256 hash of all Splunk executable files.
3. Start the Splunk (Indexer):

   As the splunk user, execute:

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

4. Perform the following activities to/on the TOE:

   - Restart the External Trusted Data Feed (Splunk Forwarder), such that it initiates a connection to the Indexer instance of the TOE.
   - Navigate and authenticate to the TOE's web UI administration URL:

     https://<TOE-IP-Address>:8000

   - Perform a query on the TOE to cause it to send an e-mail alert to the configured SMTP server:

     index=_internal | head 5 | sendmail to=example@splunk.com server=splunk2022-smtp.catl.local:25 subject="Here is an email from Splunk" message="This is an example message" use_ssl=true use_tls=false

5. Calculate and record the SHA256 hash of all Splunk executable files.
6. Compare the hash of each Splunk executable file that was calculated before exercising the application with the corresponding hash of each Splunk executable file that was calculated after exercising the application.
7. Verify that the output in Step 5 results in identical matches (i.e. empty diff file) for each corresponding executable file.

Test Results:	The evaluator verified that the TOE application executable files were not changed by the application after it has been running. The evaluator confirmed this by calculating the SHA256 hash of each TOE application executable file, exercising the functionality of the application, then re-calculated the SHA256 hash of the executable files. The evaluator confirmed that the SHA256 hashes were identical in both instances.
Execution Method:	Manual

099	FPT_TUD_EXT.2.1 – Integrity for Installation and Update
Test Objective:	The evaluator shall ensure that the application is packaged in the format of the package management infrastructure of the chosen distribution. For example, applications running on Red Hat and Red Hat derivatives shall be packaged in RPM format. Applications running on Debian and Debian derivatives shall be packaged in DEB format.
1. Inspect the TOE application package RPM file in a hex editor. 2. Verify that it has file signature “ed ab ee db”.	
Test Results:	The evaluator performed this test by inspecting the TOE application package RPM file in a hex editor and verifying that it contained the specified file signature.
Execution Method:	Manual

100	FPT_TUD_EXT.2.2 – Integrity for Installation and Update – TD0664
Test Objective:	The evaluator shall record the path of every file on the entire filesystem prior to installation of the application, and then install and run the application. Afterwards, the evaluator shall then uninstall the application, and compare the resulting filesystem to the initial record to verify that no files, other than configuration, output, and audit/log files, have been added to the filesystem.

1. Record every path of every file on the entire filesystem.
2. Install the Splunk application to the system:

    a. Install the main Splunk RPM as the root user:

    rpm -i <splunk_package_name>

    b. Move Splunk configuration files from their default location to an alternative location:

    mv /opt/splunk/etc/* /etc/opt/splunk
    rmdir /opt/splunk/etc

    c. (As root) Generate public/private keypair for the Splunk “audit.conf” functionality:

    export SPLUNK_ETC=/etc/opt/splunk
    mkdir -p $SPLUNK_ETC/auth/audit

    chown -R splunk:splunk /etc/opt/splunk
    su - splunk

    openssl ecparam -name prime256v1 -genkey -out $SPLUNK_ETC/auth/audit/private.pem -noout

    NOTE: Set the password to “password4audit”

    openssl ec -aes256 -in $SPLUNK_ETC/auth/audit/private.pem -out $SPLUNK_ETC/auth/audit/private.encrypted.pem

    openssl ec -in $SPLUNK_ETC/auth/audit/private.pem -out $SPLUNK_ETC/auth/audit/public.pem -outform PEM -pubout

    d. (As root) Generate public/private keypair for the Splunk “distServerKeys.conf” functionality:

    export SPLUNK_ETC=/etc/opt/splunk
    mkdir -p $SPLUNK_ETC/auth/distServerKeys

    chown -R splunk:splunk /etc/opt/splunk
    su - splunk

    openssl ecparam -name prime256v1 -genkey -out $SPLUNK_ETC/auth/distServerKeys/private.pem -noout

NOTE: Set the password to “password4dist”

openssl ec -aes256 -in $SPLUNK_ETC/auth/distServerKeys/private.pem -out $SPLUNK_ETC/auth/distServerKeys/private.encrypted.pem

openssl ec -in $SPLUNK_ETC/auth/distServerKeys/private.pem -out $SPLUNK_ETC/auth/distServerKeys/trusted.pem -outform PEM -pubout

e. (As root) Install the Splunk SELinux rpm:

If RHEL 8.2:

yum install splunk-selinux-0-0.9.0.el8.noarch.rpm

If RHEL 7.9:

yum install splunk-selinux-0-0.9.0.el7.noarch.rpm

3. Enable Common Criteria mode (as splunk user):

Modify / append the pre-existing /etc/opt/splunk/splunk-launch.conf file:

```
PYTHONHTTPSVERIFY=1
SPLUNK_COMMON_CRITERIA=1
SPLUNK_FIPS=1
# Do not generate python byte code
PYTHONDONTWRITEBYTECODE=1
```

4. Populate local configuration files to /etc/opt/splunk/system/local:

Untar the pre-written configuration files located in /home/splunk/configs.tar:

```
tar -xvf configs.tar
mv ./etc/opt/splunk/system/local/*.conf /etc/opt/splunk/system/local
```

5. Populate the CC-compliant certificates to /etc/opt/splunk/auth/mycerts:

Untar the pre-generated CC-compliant X.509v3 certificate files located in /home/splunk/mycerts.tar:

```
tar -xvf mycerts.tar
mkdir -p /etc/opt/splunk/auth/mycerts
mv ./etc/opt/splunk/auth/mycerts/*.pem /etc/opt/splunk/auth/mycerts
```

6. Populate the CRL files to /etc/opt/splunk/auth/crl:

Untar the pre-generated CC-compliant CRL files located in /home/splunk/crl.tar:

```
tar -xvf crl.tar
mv ./etc/opt/splunk/auth/crl/*.pem /etc/opt/splunk/auth/crl
```

7. Remove temporary directories created from the untar steps:

```
rm -rf /home/splunk/etc
```

8. Refer to FCS_STO_EXT.1 – Test Case 007:
   a. Perform Steps 1 through 6.

9. Start the Splunk application services:

As the splunk user, execute:

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

10. Perform some activities within the Splunk application:
    - Perform a query on the TOE to cause it to send an e-mail alert to the configured SMTP server:

      index=_internal | head 5 | sendmail to=example@splunk.com server=splunk2022-smtp.catl.local:25 subject="Here is an email from Splunk" message="This is an example message" use_ssl=true use_tls=false

11. Uninstall the Splunk application:
    a. Stop Splunk as the splunk user:

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

    b. Uninstall splunk binaries / data files (as the root user):

       yum remove splunk-selinux
       cp -R /etc/opt/splunk/ /opt/splunk/etc
       rm -rf /etc/opt/splunk/*
       yum remove splunk
       rm -rf /opt/splunk

    c. List and terminate the “gnome-keyring-daemon --unlock" process(es) (as the splunk user):

       ps -aux | grep gnome-keyring
       pkill gnome-keyring

    d. Remove the gnome-keyring storage files (as the root user):

       rm -rf /home/splunk/.local/share/keyrings/*

12. Record every path of every file on the entire filesystem.
13. Create a difference file to show what is left after the install compared to the original list.
14. Create a filtered list removing all of the Splunk items to determine what was modified or added to OS during install.
15. Inspect the difference files for any unexpected files.

Test Results:	The evaluator performed this test by recording the path of every file on the entire filesystem before installing the TOE application. The evaluator then installed the TOE application, launched it, exercised some of its functionality, then uninstalled the application. The evaluator then recorded the path of every file on the entire filesystem once more, compared it to the initial record, and confirmed that no files, other than configuration, output, and audit/log files were added to the filesystem.
Execution Method:	Manual

5.3.7 Trusted Path/Channel

102	**FTP_DIT_EXT.1 – Protection of Data in Transit**
Test Objective:	The evaluator shall perform the following tests. Test 1: The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The

	evaluator shall verify from the packet capture that the traffic is encrypted with HTTPS, TLS, DTLS, SSH, or IPsec in accordance with the selection in the ST.

Encryption of traffic to and from the TOE using HTTPS and TLS is tested as part of FCS_HTTPS_EXT.1, FCS_TLSC_EXT.1, and FCS_TLSS_EXT.1.

FTP_DIT_EXT.1 – Test 2 (Test Case 103) and FTP_DIT_EXT.1 – Test 3 (Test Case 104) are combined into a single test and tested in conjunction with this test because the only sensitive data transmitted to/from the TOE are administrative credentials during authentication to the Splunk web GUI.

1. Begin capturing packets using Wireshark between the TOE and the management workstation.
2. On the test system, navigate to the Splunk administration console page and attempt to authenticate using the "admin" username and "P@sswurd!" password.

   https://<TOE-IP-Address>:8000

3. Navigate to "Administrator" > "Account Settings".
4. Specify the current password ("P@sswurd!"), the new password ("catchme123"), and confirm the new password ("catchme123").
5. Click "Save".
6. Log out of the TOE web GUI.
7. Authenticate to the TOE web UI using the "admin" username and "catchme123" password.
8. Stop capturing packets using Wireshark between the TOE and the test client machine.
9. Inspect the packet capture for known plaintext credential value "catchme123".

Test Results:	The evaluator performed this test by first capturing packets between the TOE application and the management workstation. The evaluator accessed the TOE application from the management workstation, modified the authentication credentials, re-authenticated to the TOE application, and then stopped capturing packets. The evaluator examined the packet capture and did not detect any plaintext presence of the known credential data. The evaluator also confirmed that the traffic was encrypted using TLS / HTTPS.
Execution Method:	Manual

103	**FTP_DIT_EXT.1 – Protection of Data in Transit**
Test Objective:	The evaluator shall perform the following tests. Test 2: The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall review the packet capture and verify that no sensitive data is transmitted in the clear.
This test is performed as part of FTP_DIT_EXT.1 – Test Case 102.	
Test Results:	This test is performed as part of FTP_DIT_EXT.1 – Test Case 102.
Execution Method:	Manual

104	**FTP_DIT_EXT.1 – Protection of Data in Transit**
Test Objective:	The evaluator shall perform the following tests. Test 3: The evaluator shall inspect the TSS to determine if user credentials are transmitted. If credentials are transmitted the evaluator shall set the credential to a known value. The evaluator shall capture packets from the application while causing credentials to be transmitted as described in the TSS. The evaluator shall perform a string search of the captured network packets and verify that the plaintext credential previously set by the evaluator is not found.
This test is performed as part of FTP_DIT_EXT.1 – Test Case 102.	
Test Results:	This test is performed as part of FTP_DIT_EXT.1 – Test Case 102.
Execution Method:	Manual

5.3.8 Vulnerability Testing

The evaluation team reviewed vendor documentation, formulated hypotheses, performed vulnerability analysis, and documented the hypotheses and analysis in accordance with the APP PP requirements. Keywords were identified based upon review of the Security Target and AGD. The following keywords were identified:

Keyword	Description
Splunk	This is a generic term for searching for known vulnerabilities produced by the company as a whole.
Splunk Enterprise (Version 9.0)	This is a generic term for searching for known vulnerabilities for the specific product. In this case Splunk would find the vulnerability and Enterprise would be used to narrow the list to a specific software product and version.
Libraries	
OpenSSL(1.0.2zf-fips)	Provides all of the security encryptions functionality required by Splunk
Declared library list from FPT_LIB_EXT.1	

These keywords were used individually and as part of various permutations and combinations to search for vulnerabilities on public vulnerability sources (updated March 15, 2023). The following public vulnerability sources were searched:

- NIST National Vulnerabilities: https://web.nvd.nist.gov/view/vuln/search
- Common Vulnerabilities and Exposures: http://cve.mitre.org/cve/ https://www.cvedetails.com/vulnerability-search.php
- US-CERT: http://www.kb.cert.org/vuls/html/search
- Tipping Point Zero Day Initiative http://www.zerodayinitiative.com/advisories
- Offensive Security Exploit Database: https://www.exploit-db.com/

Upon the completion of the vulnerability analysis research, the team had identified several generic vulnerabilities upon which to build a test suite. These tests were created specifically with the intent of exploiting these vulnerabilities within the TOE or its configuration.

The team tested the following areas:

Therefore, the team tested the following area:

- Burp Suite Scan
  - Perform security testing of web applications
- Virus/Malware Scan
  - Perform a virus scan on software as required by the AppPP assurance activity requirements.

As a result of vulnerabilities found during the public search, the TOE's version for evaluation was upgraded to 9.0.4. The lab evaluated the Release notes for versions 9.0.2, 9.0.3, and 9.0.4 to assess any impact to testing. The six updates identified in 9.0.2 and 9.0.3 were part of non-TSF related items and have no impact on the evaluation. The update for 9.0.4 contained only fixes to mitigate the recently reported vulnerabilities on the web-UI.

The lab found no open vulnerabilities for the current version of the TOE (9.0.4) between the public search and performing the virus scan and the BurpSuite scan.

Verdict: The evaluation team has completed testing of this component, resulting in a verdict of PASS.

6 Conclusions

The TOE was evaluated against the ST and has been found by this evaluation team to be conformant with the ST. The overall verdict for this evaluation is: Pass.

7 Glossary of Terms

Acronym	Definition
AES	Advanced Encryption Standard
ASLR	Address Space Layout Randomization
CA	Certification Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CLI	Command Line Interface
CRL	Certificate Revocation List
CSP	Critical Security Parameter
DHE	Diffie-Hellman Key Exchange
DRBG	Deterministic Random Bit Generator
ECDHE	Elliptic Curve Diffie-Hellman Key Exchange
ECDSA	Elliptic Curve Digital Signature Algorithm
GCM	Galois/Counter Mode
HMAC	Hashed Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IT	Information Technology
JIT	Just-in-Time (compilation)
OS	Operating System
OSP	Organizational Security Policy
PCRE	Perl Compatible Regular Expressions
PP	Protection Profile
NIAP	National Information Assurance Partnership
RA	Registration Authority
RBG	Random Bit Generator
RHEL	Red Hat Enterprise Linux
SAR	Security Assurance Requirement
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SMTP	Simple Mail Transfer Protocol
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
UI	User Interface

Table 6-1: Acronyms

Terminology	Definition
Authorized Administrator	Any user which has been assigned to a privilege level that is permitted to perform all TSF-related functions.
Role	An assigned role gives a user varying access to the management of the TOE.
Security Administrator	Synonymous with Authorized Administrator for the purposes of this evaluation.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Table 6-2: Terminology