Adtran's FSP 3000R7 Network Element r22.2.2

Assurance Activities Report

Version 1.0
March 17, 2024

Evaluated by:

Booz | Allen | Hamilton®

Common Criteria Test Laboratory
NVLAP Lab Code # 200423
1100 West Street
Laurel, MD 20707

Evaluation Personnel:
Herbert Markle
Christopher Rakaczky
Evan Seiz

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

TOE Developer and Evaluation Sponsor:

Adtran Networks North America, Inc.
(formerly known as ADVA Optical Networking North America, Inc)
5755 Peachtree Industrial Boulevard
Norcross, Georgia 30092

The Author of the Security Target:

Booz | Allen | Hamilton®

Common Criteria Test Laboratory
NVLAP Lab Code # 200423
1100 West Street
Laurel, MD 20707

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

Purpose

The purpose of this document is to serve as a non-proprietary attestation that this evaluation has satisfied all of the TSS, AGD, ATE and AVA Assurance Activities required by the Protection Profiles/Extended Packages to which the TOE claims exact conformance.

1 TOE Summary Specification Assurance Activities

The evaluation team completed the testing of the Security Target (ST) '*Adtran's FSP 3000R7 Network Element r22.2.2 Security Target'* and confirmed that the TOE Summary Specification (TSS) contains all Assurance Activities as specified by the 'Collaborative Protection Profile for Network Devices Version 2.2e' (NDcPP). 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 NDcPP Assurance Activities.

Through the evaluation of ASE_TSS.1-1, described in the ETR, the evaluators were able to determine that each SFR was described in enough detail to demonstrate that the TSF addresses the SFR. 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 NDcPP that defines where the most up-to-date TSS Assurance Activity was defined.

Note: The TOE is a standalone TOE. Therefore, responses to assurance activities for distributed TOEs have been omitted for clarity.

FAU_GEN.1 – *"For the administrative task of generating/import of, changing, or deleting of cryptographic keys as defined in FAU_GEN.1.1c, the TSS should identify what information is logged to identify the relevant key.*

For distributed TOEs the evaluator shall examine the TSS to ensure that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components. The evaluator shall ensure that this mapping of audit events to TOE components accounts for, and is consistent with, information provided in Table 1, as well as events in Tables 2, 4, and 5 (where applicable to the overall TOE). This includes that the evaluator shall confirm that all components defined as generating audit information for a particular SFR should also contribute to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component cover all the SFRs that it implements."

Section 8.1.1 of the ST includes an example audit record for importing a certificate used for public key authentication for SSH access. The audit record includes: Timestamp, user

importing, add key event with certificate details including: purpose, key algorithm, key length and fingerprint of certificate.

This activity passes as the description provides the required identification on what information is logged to identify the relevant key for administrative key management.

FAU_GEN.2 – *"The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1."*

FAU_STG_EXT.1 – *"The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.*

The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access.

The evaluator shall examine the TSS to ensure it describes whether the TOE is a standalone TOE that stores audit data locally or a distributed TOE that stores audit data locally on each TOE component or a distributed TOE that contains TOE components that cannot store audit data locally on themselves but need to transfer audit data to other TOE components that can store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs it contains a list of TOE components that store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs that contain components which do not store audit data locally but transmit their generated audit data to other components it contains a mapping between the transmitting and storing TOE components.

The evaluator shall examine the TSS to ensure that it details the behaviour of the TOE when the storage space for audit data is full. When the option 'overwrite previous audit record' is selected this description should include an outline of the rule for overwriting audit data. If 'other actions' are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.

The evaluator shall examine the TSS to ensure that it details whether the transmission of audit information to an external IT entity can be done in real time or periodically. In case the TOE does not perform transmission in real time the evaluator needs to verify that the TSS provides details about what event stimulates the transmission to be made as well as the possible acceptable frequency for the transfer of audit data.

For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how audit data transfer to the external audit server is implemented among the different TOE components (e.g. every TOE components does its own transfer or the data is sent to another TOE component for central transfer of all audit events to the external audit server).

For distributed TOEs the evaluator shall examine the TSS to ensure it describes which TOE components are storing audit information locally and which components are buffering audit information and forwarding the information to another TOE component for local storage. For every component the TSS shall describe the behaviour when local storage space or buffer space is exhausted."

Section 8.1.2 of the ST states the TOE is a standalone appliance responsible for storing and sending its own audit records. The TOE automatically forwards audit records to an external audit server via TLS in near real-time. The TSS states that the TOE's audit logs take up a total of 4.8MB. The TOE uses a FIFO methodology when rolling over historical audit logs to maintain the maximum storage threshold. The audit log files can be accessed at the OS level by a Security Administrator that has the ability to escalate to root privileges, using the sudo command, to make authorized file deletions or modifications.

This activity passes as the description includes the required information for audit storage, storage max size, behavior when audit is full, and remote storage are present.

FCS_CKM.1 – *"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."*

Section 8.2.1 of the ST states The TOE implements a FIPS PUB 186-4 conformant ECC key generation mechanism for establishing TLS connections. Specifically, the TOE's implementation of ECC key generation complies with FIPS 186-4 (Digital Signature Standard (DSS) Appendix B.4) supporting a 384-bit key size. Additionally, the TOE supports FFC key generation complies with NIST Special Publication 800-56A Revision 3 and RFC 3526 supporting a key size of 1024 bits.

This activity passes as the description includes both schemes used are identified and key sizes are specified.

FCS_CKM.2 – TD0580 – *"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. It is sufficient to provide the scheme, SFR, and service in the TSS.*

The intent of this activity is to be able to identify the scheme being used by each service. This would mean, for example, one way to document scheme usage could be:

Scheme	*SFR*	*Service*
RSA	*FCS_TLSS_EXT.1*	*Administration*
ECDH	*FCS_SSHC_EXT.1*	*Audit Server*
ECDH	*FCS_IPSEC_EXT.1*	*Authentication Server*

The information provided in the example above does not necessarily have to be included as a table but can be presented in other ways as long as the necessary data is available."

Section 8.2.2 of the ST states The Elliptic curve-based key establishment is used for TLS communications for remote administration using the Web GUI and exporting audit data to the Audit Server (FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1). Additionally, the TOE supports FFC based key establishment using safe prime groups in support of the TOE's SSH server service (FCS_SSHS_EXT.1).

This activity passes as the description includes all key schemes are identified and their use are specified.

FCS_CKM.4 – *"The evaluator examines the TSS to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations (e.g. factory reset or device wipe function, disconnection of trusted channels, key change as part of a secure channel protocol), and the destruction method used in each case. For the purpose of this Evaluation Activity the relevant keys are those keys that are relied upon to support any of the SFRs in the Security Target. The evaluator confirms that the description of keys and storage locations is consistent with the functions carried out by the TOE (e.g. that all keys for the TOE-specific secure channels and protocols, or that support FPT_APW.EXT.1 and FPT_SKP_EXT.1, are accounted for). In particular, if a TOE claims not to store plaintext keys in non-volatile memory then the evaluator checks that this is consistent with the operation of the TOE.*

The evaluator shall check to ensure the TSS identifies how the TOE destroys keys stored as plaintext in non-volatile memory, and that the description includes identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).

Note that where selections involve 'destruction of reference' (for volatile memory) or 'invocation of an interface' (for non-volatile memory) then the relevant interface definition is examined by the evaluator to ensure that the interface supports the selection(s) and description in the TSS. In the case of non-volatile memory, the evaluator includes in their examination the relevant interface description for each media type on which plaintext keys are stored. The presence of OS-level and storage device-level swap and cache files is not examined in the current version of the Evaluation Activity.
Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator shall check that the TSS identifies the encryption method and the key-encrypting-key used, and that the key-encrypting-key is either itself stored in an encrypted form or that it is destroyed by a method included under FCS_CKM.4.

The evaluator shall check that the TSS identifies any configurations or circumstances that may not conform to the key destruction requirement (see further discussion in the Guidance Documentation section below). Note that reference may be made to the

Guidance Documentation for description of the detail of such cases where destruction may be prevented or delayed.

Where the ST specifies the use of "a value that does not contain any CSP" to overwrite keys, the evaluator examines the TSS to ensure that it describes how that pattern is obtained and used, and that this justifies the claim that the pattern does not contain any CSPs."

Section 8.2.3 of the ST displays the table that describes the keys, origin of the keys, where the keys are stored (RAM or filesystem), and how they are destroyed. There are no known instances where key destruction does not happen as defined. The keys defined are consistent with the TLSC, TLSS, and SSH communications defined within the ST. The storage locations identified for the keys are also consistent with functionality found during the evaluation. The table describes how each key is destroyed and by what mechanism. In the case of volatile memory the table identifies the API called for the destruction. For non-volatile memory the TOE describes when the file is destroyed such as overwriting or using the rm command.

The activity passes as the description includes all keys are defined, origins specified, storage identified, and destruction method explained.

FCS_COP.1/DataEncryption – *"The evaluator shall examine the TSS to ensure it identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption."*

Section 8.2.4 of the ST identifies The TOE performs encryption and decryption using the AES algorithm in CTR and GCM modes with key sizes of 256 bits. The AES algorithm meets ISO 18033-3, CTR meets ISO 10116, and GCM meets ISO 19772. The TOE's AES implementation is validated under CAVP.

This activity passes as the description includes the key, key size, and key mode used for data encryption/decryption are identified.

FCS_COP.1/SigGen – *"The evaluator shall examine the TSS to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services."*

Section 8.2.5 of the ST specifies The TOE performs digital signature services generation and verification in accordance with Elliptic Curve Digital Signature Algorithm (ECDSA) with key sizes (modulus) 384 bits.

This activity passes as the description includes the claimed key and key size used for signature services is identified.

FCS_COP.1/Hash – *"The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS."*

Section 8.2.6 of the ST states that:

- SHA-384 for TLS (FCS_TLSC_EXT.1/ FCS_TLSS_EXT.1)
- SHA-384 for TLS NIST curves (FCS_TLSC_EXT.1/ FCS_TLSS_EXT.1)
- SHA-384 for HMAC (FCS_COP.1/KeyedHash)
- SHA-384 for software integrity check (FPT_TST_EXT.1)
- SHA-384 for NTP timestamp verification (FCS_NTP_EXT.1)
- SHA-384 for trusted update digital signature verification (FPT_TUD_EXT.1)
- SHA-512 for password hashing (FPT_APW_EXT.1)

The above bullets are consistent with the claims identified within the ST.

This activity passes as the description includes all the hash functions supported and maps the hash to the specific usage.

FCS_COP.1/KeyedHash – *"The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used."*

Section 8.2.7 of the ST states that HMAC-SHA-384 [key-size: 384 bits, digest size: 384 bits, block size: 1024 bits, MAC lengths: 384 bits] for TLS communication support.

This activity passes as the description includes all the keyed hash functions with their respective key size, digest size, block size, and MAC length.

FCS_NTP_EXT.1.1 – *"The evaluator shall examine the TSS to ensure it identifies the version of NTP supported, how it is implemented and what approach the TOE uses to ensure the timestamp it receives from an NTP timeserver (or NTP peer) is from an authenticated source and the integrity of the time has been maintained.*

The TOE must support at least one of the methods or may use multiple methods, as specified in the SFR element 1.2. The evaluator shall ensure that each method selected in the ST is described in the TSS, including the version of NTP supported in element 1.1, the message digest algorithms used to verify the authenticity of the timestamp and/or the protocols used to ensure integrity of the timestamp."

Section 8.2.8 of the ST states the TOE only support NTP v4 in accordance with RFC 5905. The system time is updated via NTP client-server authentication. The TOE uses SHA-384 message digest algorithm to verify the authenticity of the timestamp which ensures reliability. The TOE supports a maximum of 3 NTP servers. The TOE will not update NTP timestamp from broadcast and/or multicast addresses.

The activity passes as the description includes that NTP v4 is the only mechanism supported for NTP and identifies the use of SHA-384 message digest algorithm to verify the authenticity of the timestamp to ensure reliability.

FCS_RBG_EXT.1 – *"The evaluator shall examine the TSS to determine that it specifies the DRBG type, identifies the entropy source(s) seeding the DRBG, and state the assumed or calculated min-entropy supplied either separately by each source or the min-entropy contained in the combined seed value."*

Section 8.2.9 of the ST states that the TOE uses the CTR_DRBG and cannot be changed. The DRBG is seeded with a minimum of 256-bit security strength. The TOE relies on kernel modules (software) to gather and output entropy for the TOE's random requirements. Additionally, the TOE uses a hardware source to produce entropy to fill the entropy pool quicker during the boot process. This hardware source is not used during operational runtime. The entropy pools are protected by being in kernel memory and are not accessible from user space. The entropy source is described in greater detail in the proprietary Entropy Assessment Report.

The activity passes as the description includes the DRBG type and entropy source seeding the DRBG. The calculated min-entropy is supplied separately in a proprietary Entropy Assessment Report.

FCS_HTTPS_EXT.1 – *"The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818."*

Section 8.2.10 of the ST explains that the TOE uses an HTTPS implementation that conforms to RFC 2818 and uses the TLS server implementations that covers the functionality specified in FCS_TLSS_EXT.1 and FCS_TLSS_EXT.2 (mutual authentication).

The activity passes as the description shows compliance to RFC 2818 TLS server.

FCS_SSHS_EXT.1.1 – This SFR does not contain any NDcPP TSS Assurance Activities.

FCS_SSHS_EXT.1.2 – **TD0631** – *"The evaluator shall check to ensure that the TSS contains a list of supported public key algorithms that are accepted for client authentication and that this list is consistent with signature verification algorithms selected in FCS_COP.1/SigGen (e.g., accepting EC keys requires corresponding Elliptic Curve Digital Signature algorithm claims).*

The evaluator shall confirm that the TSS includes the description of how the TOE establishes a user identity when an SSH client presents a public key or X.509v3 certificate. For example, the TOE could verify that the SSH client's presented public key matches one that is stored within the SSH server's authorized_keys file.

If password-based authentication method has been selected in the FCS_SSHS_EXT.1.2, then the evaluator shall confirm its role in the authentication process is described in the TSS."

Section 8.2.11 of the ST specifies that the TOE's implementation of SSHv2 only supports ecdsa-sha2-nistp384 for public key algorithm (user and host). If a public key is presented for user authentication, the TOE will verify that the SSH client's presented public key matches one that is stored within the SSH server's authorized keys database. . In the case of password-based authentication attempt, the presented user credentials are verified using the TOE's native password authentication mechanism.

This activity passes as the description includes the identification of the supported public key algorithm accepted for client authentication, how the TOE establishes user identity for public key authentication, and the use of password-based authentication methods. This is consistent with Section 6 of the ST.

FCS_SSHS_EXT.1.3 – *"The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled."*

Section 8.2.11 of the ST states the TOE's SSH implementation will detect all large packets greater than 32,768 bytes and drop accordingly.

This activity passes as the description includes the handling process of when the TSF receives a large packet and identifies the size of a large packet.

FCS_SSHS_EXT.1.4 – *"The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component."*

Section 8.2.11 of the ST specifies that the TOE's implementation of SSHv2 only supports aes256-gcm@openssh.com as the encryption algorithm.

This activity passes as the encryption algorithm is identified and is identical to the defined encryption algorithm in Section 6 of the ST.

FCS_SSHS_EXT.1.5 – **TD0631** – *"The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the SSH server's host public key algorithms supported are specified and that they are identical to those listed for this component."*

Section 8.2.11 of the ST specifies that the TOE's implementation of SSHv2 only supports ecdsa-sha2-nistp384 for public key algorithm (user and host).

This activity passes as the host public key algorithm is identified and is identical to the defined host public key algorithm in Section 6 of the ST.

FCS_SSHS_EXT.1.6 – *"The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that the list corresponds to the list in this component."*

Section 8.2.11 of the ST lists that the TOE's implementation of SSHv2 only supports the MAC algorithm of "implicit" due to the selection of aes256-gcm@openssh.com for encryption algorithm.

This activity passes as the MAC algorithm is identified and is identical to the defined MAC algorithm in Section 6 of the ST.

FCS_SSHS_EXT.1.7 – *"The evaluator shall check the TSS to ensure that it lists the supported key exchange algorithms, and that the list corresponds to the list in this component."*

Section 8.2.11 of the ST lists that the TOE's implementation of SSHv2 only supports diffie-hellman-group15-sha512 and ecdh-sha2-nistp384 for key exchange methods.

This activity passes as the key exchange algorithms are identified and are identical to the defined key exchange algorithm in Section 6 of the ST.

FCS_SSHS_EXT.1.8 – *"The evaluator shall check that the TSS specifies the following:*

a) Both thresholds are checked by the TOE.

b) Rekeying is performed upon reaching the threshold that is hit first."

Section 8.2.11 of the ST states the TSF enforces the connection to be rekeyed after no longer than one hour, and no more than one gigabyte of transmitted data, whichever threshold is reached first. The SSH rekey time and size threshold parameters are not administratively configurable.

The activity passes as the description specifies both time and amount of transmitted data as thresholds that cause a rekey event and that it is whichever threshold is met first that triggers the rekey event.

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

Section 8.2.12 of the ST states that the TOE's TLSv1.2 client implementation only supports the TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

This activity passes as the description includes the identification of protocol version used and supported cipher. These are identical to the defined key exchange algorithm in Section 6 of the ST.

FCS_TLSC_EXT.1.2 – *"The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the administrator/applicationconfigured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.*

Note that where a TLS channel is being used between components of a distributed TOE for FPT_ITT.1, the requirements to have the reference identifier established by the user are relaxed and the identifier may also be established through a "Gatekeeper" discovery process. The TSS should describe the discovery process and highlight how the reference identifier is supplied to the "joining" component. Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1 and the ST author selected attributes from RFC 5280, the evaluator shall ensure the TSS describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The evaluator shall ensure the TSS presents an argument how the attribute type, or combination of attribute types, uniquely identify the remote TOE component; and the evaluator shall verify the attribute type, or combination of attribute types, is sufficient to support unique identification of the maximum supported number of TOE components.

If IP addresses are supported in the CN as reference identifiers, the evaluator shall ensure that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order. The evaluator shall also ensure that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced."

Section 8.2.12 of the ST states the TOE, upon the presentation of the X.509v3 server host certificate, will validate the certificate per FIA_X509_EXT.1/REV requirements. The TSF shall verify that the presented identifier matches the reference identifier (IPv4 address in CN or SAN) in the certificate. The Common Name and Subject Alternative Name (IPv4 address only) are the only reference identifiers in the certificate that are part of that validation.

In the evaluated configuration, the TOE only supports Common Name (CN) and Subject Alternative Name (SAN) reference identifiers that are using IPv4 address values. Canonical formatting according to RFC 3986 is enforced. The TOE does not support the use of IPv6 addresses, URI, DNS (FQDN), service name reference identifiers, wildcards or pinned certificates.

The TSF converts that IP address, obtained from the certificate, from ASN.1 to the binary representation of the textual string of the IP address. The TSF also converts the IP address from the established network connection to the binary representation of the textual string of the IP address. The two representations are then compared to determine what action is performed next. The methodology for performing the check is as follows:

- If the SAN value exists:
  - If the two values match, revocation checking using the CRL is performed.
  - If the two values do not match, the certificate is deemed invalid and the connection is immediately terminated.
- If the SAN field is not used (non-existent), the representation of the CN value is used for comparison instead:
  - If the two values match, revocation checking using the CRL is performed.
  - If the two values do not match, the certificate is deemed invalid and the connection is immediately terminated.

This activity passes as the description includes the identification of what reference identifiers are supported and provides a description on how the IPv4 address is parsed (searched) from the certificate for verification and the enforcement of canonical formatting according to RFC 3986.

This activity passes as the description includes the identification of what reference identifiers are supported and provides a description on how the IPv4 address is parsed (searched) from the certificate for verification and the enforcement of canonical formatting according to RFC 3986.

FCS_TLSC_EXT.1.3 – This SFR does not contain any NDcPP TSS Assurance Activities.

FCS_TLSC_EXT.1.4 – *"The evaluator shall verify that TSS describes the Supported Elliptic Curves/Supported Groups Extension and whether the required behaviour is performed by default or may be configured."*

Section 8.2.12 of the ST states the TOE's TLSv1.2 implementation only supports the secp384r1 Elliptic Curves when placed in its evaluated configuration and shall present the secp384r1 curve in the Supported Elliptic Curves/Supported Groups Extensions of the Client Hello.

This activity passes as the description includes the identification of the Elliptic Curves/Supported Groups Extension is supported and defines that, in the evaluated configuration, secp384r1 is the only curve supported.

FCS_TLSS_EXT.1.1 – *"The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The*

evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component."

Section 8.2.13 of the ST states that the TOE's TLSv1.2 server implementation only supports the TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384.

This activity passes as the description includes the identification of protocol version used and supported cipher. These are identical to the defined key exchange algorithm in Section 6 of the ST.

FCS_TLSS_EXT.1.2 – "*The evaluator shall verify that the TSS contains a description of how the TOE technically prevents the use of old SSL and TLS versions."*

Section 8.2.13 of the ST states the TOE will deny connections from a client requesting any protocol versions besides TLS v1.2. When the TOE receives a TLS connection request with the wrong (unsupported) version, it returns a Fatal Alert: Handshake failure message and terminates the connection.

The activity passes as the description states that the wrong TLS version will cause the TOE to terminate the connection with a failure notice.

FCS_TLSS_EXT.1.3 – **TD0635** *"If using ECDHE and/or DHE ciphers, the evaluator shall verify that the TSS lists all EC Diffie-Hellman curves and/or Diffie-Hellman groups used in the key establishment by the TOE when acting as a TLS Server. For example, if the TOE supports TLS_DHE_RSA_WITH_AES_128_CBC_SHA cipher and Diffie-Hellman parameters with size 2048 bits, then list Diffie-Hellman Group 14."*

Section 8.2.13 of the ST states that secp384r1 is the only Elliptic Curves supported.

This activity passes as the required information was found.

FCS_TLSS_EXT.1.4 – TD0569 *"The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246) and/or if session resumption based on session tickets is supported (RFC 5077).*

If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption. The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.

If session tickets are supported, the evaluator shall verify that the TSS describes that session tickets adhere to the structural format provided in section 4 of RFC 5077 and if not, a justification shall be given of the actual session ticket format.

If the TOE claims a (D)TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator verifies that the TSS describes how session

resumption operates (i.e. what would trigger a full handshake, e.g. checking session status, checking Session ID, etc.). If multiple contexts are used the TSS describes how session resumption is coordinated across those contexts. In case session establishment and session resumption are always using a separate context, the TSS shall describe how the contexts interact with respect to session resumption (in particular regarding the session ID). It is acceptable for sessions established in one context to be resumable in another context."

Section 8.2.13 of the ST states that neither session tickets nor session resumption is supported.

This activity passes as TOE does not support session tickets nor session resumption.

FCS_TLSS_EXT.2.1 and 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.*

The evaluator shall verify the TSS describes how the TSF uses certificates to authenticate the TLS client. The evaluator shall verify the TSS describes if the TSF supports any fallback authentication functions (e.g. username/password, challenge response) the TSF uses to authenticate TLS clients that do not present a certificate. If fallback authentication functions are supported, the evaluator shall verify the TSS describes whether the fallback authentication functions can be disabled."

Section 8.2.14 of the ST states the TOE, upon the presentation of the X.509v3 client side certificate, will validate the certificate per FIA_X509_EXT.1/REV requirements when mutual authentication has been configured.

In the evaluated configuration, the TOE only supports Common Name (CN) and Subject Alternative Name (SAN) reference identifiers that are using IPv4 address values. Canonical formatting according to RFC 3986 is enforced. The TOE does not support the use of IPv6 addresses, URI, DNS (FQDN), service name reference identifiers, wildcards or pinned certificates.

Additionally, there is no fallback authentication fallback position if the certificate validation fails. There is no administrative override mechanism to force the connection if the peer certificate is deemed invalid.

This activity passes as the description includes the required use of X.509v3 certificates the TOE validates the certificate according to the FIA_X509_EXT.1/REV requirements, and there is no fallback position claimed.

FCS_TLSS_EXT.2.3 – *"The evaluator shall verify that the TSS describes which types of identifiers are supported during client authentication (e.g. Fully Qualified Domain Name (FQDN)). If FQDNs are supported, the evaluator shall verify that the TSS describes that corresponding identifiers are matched according to RFC6125. For all other types of*

identifiers, the evaluator shall verify that the TSS describes how these identifiers are parsed from the certificate, what the expected identifiers are and how the parsed identifiers from the certificate are matched against the expected identifiers."

Section 8.2.14 of the ST states that the Common Name and Subject Alternative Name (IPv4 address only) are the only reference identifiers in the certificate that are part of that validation. The TOE does not support URI, DNS (FQDN), service name reference identifiers, wildcards or pinned certificates. The TOE does not support URI, DNS (FQDN), service name reference identifiers, wildcards or pinned certificates.

In the evaluated configuration, the TOE only supports Common Name (CN) and Subject Alternative Name (SAN) reference identifiers that are using IPv4 address values. Canonical formatting according to RFC 3986 is enforced. The TOE does not support the use of IPv6 addresses, URI, DNS (FQDN), service name reference identifiers, wildcards or pinned certificates.

The TSF converts that IP address, obtained from the certificate, from ASN.1 to the binary representation of the textual string of the IP address. The TSF also converts the IP address from the established network connection to the binary representation of the textual string of the IP address. The two representations are then compared to determine what action is performed next. The methodology for performing the check is as follows:

- If the SAN value exists:
    - If the two values match, revocation checking using the CRL is performed.
    - If the two values do not match, the certificate is deemed invalid and the connection is immediately terminated.
- If the SAN field is not used (non-existent), the representation of the CN value is used for comparison instead:
    - If the two values match, revocation checking using the CRL is performed.
    - If the two values do not match, the certificate is deemed invalid and the connection is immediately terminated.

This activity passes as the description describes that IPv4 addresses are the only allowable entry for the CN and SAN which are the only valid reference identifiers claimed and a description of how the IP address is parsed for comparison.

FIA_AFL.1 – *"The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.*

The evaluator shall examine the TSS to confirm that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking)."

Section 8.3.1 of the TSS states the TSF provides a administratively configurable counter threshold for consecutive failed password authentication attempts that will lock a user account for a defined period of time when the failure counter threshold is reached. The failure threshold counter is configured on a per account basis with a value of 1-10.

A single failure counter is used per user across all interfaces (local, SSH, HTTPS). The failure counter increases with every failed login attempt, regardless of which interface is used, until the counter reaches its administratively defined threshold. A successful password-based authentication occurring, through any interface, prior to the failure counter reaching its threshold will reset the failure counter to 0.

The user account will automatically unlock after the configured time interval has passed. The Security Administrator can configure the lockout period between 0-99999 seconds. Alternatively, an administrative account from any interface has the ability to unlock another administrative account in the event of an administrative account reaching the failed authentication attempts threshold.

For the evaluated configuration, the serial Access Lockout setting must be set to "Do Not Lock Admins" so the TOE does not lock the administrator role accounts on the serial physical interface (local access) but does lock the accounts from remote access.

This activity passes as the description describes the lock out functionality, how the administrator or TSF will unlock the account can manually unlock the account, how the account will unlock at a time period configured by an administrator, and how the TOE must be configured so there is never a situation where no administrator access is available, either permanently or temporarily.

FIA_PMG_EXT.1 – **TD0792** *"The evaluator shall check that the TSS lists the supported special character(s)for the composition of administrator passwords.*

The evaluator shall check the TSS to ensure that the minimum_password_length parameter is configurable by a Security Administrator.

The evaluator shall check that the TSS lists the range of values supported for the minimum_password_length parameter. The listed range shall include the value of 15."

Section 8.3.2 of the ST states the TOE accepted special characters include: "!", "@", "#", "$", "%", "^", "(", ")", "_" , "+" , "|" , "~" , "{", " }" , "[" , "]" , "-", ".". Additionally, the TOE supports the ability for a Security Administrator to set the minimum password length to 15 characters or greater with a maximum of 128 characters via any administrative interface.

This activity passes as the special character supported are listed and the defined range supports a 15 character password length.

FIA_UAU.7 – This SFR does not contain any NDcPP TSS Assurance Activities.

FIA_UAU_EXT.2 – *"Evaluation Activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1."*

FIA_UIA_EXT.1 – *"The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon".*

The evaluator shall examine the TSS to determine that it describes which actions are allowed before user identification and authentication. The description shall cover authentication and identification for local and remote TOE administration.

For distributed TOEs the evaluator shall examine that the TSS details how Security Administrators are authenticated and identified by all TOE components. If not, all TOE components support authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the TSS shall describe how the overall TOE functionality is split between TOE components including how it is ensured that no unauthorized access to any TOE component can occur.

For distributed TOEs, the evaluator shall examine the TSS to determine that it describes for each TOE component which actions are allowed before user identification and authentication. The description shall cover authentication and identification for local and remote TOE administration. For each TOE component that does not support authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 the TSS shall describe any unauthenticated services/services that are supported by the component."

Section 8.3.4 of the ST states the display and acknowledgement of a warning banner is the only TOE functionality that is available to an unauthenticated user of the Web GUI, Remote CLI, and Local CLI.

This section also identifies that there are 3 ways to logon to the TOE: SSH to remotely access the CLI, HTTPS to remotely access the web GUI, and locally to access the CLI. All methods accept username/password credentials to authenticate to the TOE which in turn validates the credentials using a local mechanism. For an authentication request using an SSH connection, the user can provide a public-key instead of the username/password combination. The TSF will validate the public-key against the administratively imported and internally stored public-key assigned to that user requesting access. In all cases, only a successful validation of the presented credentials/public-key provides access to the administrative interfaces of the TOE.

This activity passes as the description covers the functionality provided before logon, identifies all methods of authenticating to the TOE, identifies the different credential types allowed, and defines a what constitutes a successful logon.

FIA_X509_EXT.1/Rev – *"The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). It is expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected). It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).*

The TSS shall describe when revocation checking is performed and on what certificates. If the revocation checking during authentication is handled differently depending on whether a full certificate chain or only a leaf certificate is being presented, any differences must be summarized in the TSS section and explained in the Guidance."

Section 8.3.5 of the TSS states the TOE uses X.509v3 certificates to support authentication for TLS connections to external IT entities in accordance with RFC 5280. The TOE performs certificate validity checking for any X.509v3 certificates presented to the TOE as part of TLS connections between itself and a remote audit server (audit log transmission) or HTTPS client (remote web UI administration) with mutual authentication enabled. The TOE also validates any X.509v3 certificate used to sign a software update during the software update process.

The certificate validation steps, including the description of the extendedKeyUsage fields, are also identified in this section. These steps and descriptions are identical to the SFR definition.

This activity passes as the description covers when the certificate checks happen, identifies the rules for extendedKeyUsage fields, meets the expectation of when revocation checking is performed and on what certificates.

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 behaviour 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 guidance documentation contains instructions on how this configuration action is performed."

Section 8.3.5 of the ST states that in order for the TOE to authenticate to the external Audit Server, and support mutual authentication with the administrative workstation to access the Web GUI, the trusted CA certificates must be individually installed into the TOE's certificate trust store. The necessary instructions for importing the audit server certificate, the client certificate from the administrative workstation used for mutual authentication for Web GUI access, and installing the TOE's own server certificate are included in the AGD.

If a presented certificate is deemed valid according to FIA_X509_EXT.1/Rev, then the TSF performs certificate revocation checking according to the following rules. These rules continue to apply when the TSF cannot establish a connection to download a new CRL:

- accept the certificate if the cached CRL is not yet expired and none of the certificates in the certificate chain (including the leaf certificate) are revoked.
- reject the certificate if the cached CRL is not yet expired and if the CRL identifies that any of the certificates in the certificate chain (including the leaf) are revoked. In this case, the TSF produces an audit record that reports an error message identifying the revoked certificate.
- reject the certificate if the cached CRL is expired regardless of the TOE's ability to successfully download a newer CRL from the CRL distribution point (CDP). In this case, the TSF produces an audit record that reports an error message identifying the certificate as invalid due to an expired CRL.
- The TSF does not provide a mechanism to override the validation decision.

An expired CRL does not automatically trigger a download of a new CRL. The CRL is updated according to the frequency defined by the administrator or via a manual update by the Security Administrator. The TSF follows the above rules for determining the revocation status of a certificate chain regardless of the TOE's ability to connect to the CDP.

Additionally, the sections states that when the TSF cannot establish a connection to the CDP, the TOE will automatically continue attempting to download a new CRL at regular intervals until successful.

This activity passes as the description covers how the TOE chooses which certificates to user (installing into certificate store for a particular purpose), behavior of revocation checking including expected behavior when certificate distribution point is not available, and the guidance has the instructions for performing the required configuration.

FIA_X509_EXT.3 – *"If the ST author selects "device-specific information", the evaluator shall verify that the TSS contains a description of the device-specific fields used in certificate requests."*

This activity is trivial passed as the TOE does not claim the *"device-specific information"* selection which is consistent with the write-up in Section 8.3.5 of the ST.

FMT_MOF.1/ManualUpdate – "For distributed TOEs see chapter 2.4.1.1. There are no specific requirements for non-distributed TOEs."

This activity is trivial passed as the TOE is not distributed.

FMT_MTD.1/CoreData – *"The evaluator shall examine the TSS to determine that, for each administrative function identified in the guidance documentation; those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.*

If the TOE supports handling of X.509v3 certificates and implements a trust store, the evaluator shall examine the TSS to determine that it contains sufficient information to describe how the ability to manage the TOE's trust store is restricted."

Section 8.4.1 of the ST states that the display and acknowledgement of a warning banner is the only TOE functionality available prior to identification and authentication. The description states that the TOE utilizes role-based access control (RBAC) to restrict access to the administrative functions that manage the TSF data. The TOE limits the presented functionality based on the privileges bound to the authenticated user. The available functionality presented to an authenticated user is based on the group of permissions and the privileges associated with the permissions aligned to the authenticated user's assigned role. These permissions/privileges are bound to the user only after the user has successfully authenticated. The TSF restricts the ability to manage the TSF data to only Security Administrators. This description also includes a table which identifies the user role that is allowed to perform that task. All tasks related to X.509v3 certificate management tasks are shown to be limited to only the security administrator role.

This activity passes as the description describes that only the displaying and acknowledgment of the warning banner is available prior to authentication. The table also defines that all X.509v3 certificate management related functions are limited to just security administrators.

FMT_MTD.1/CryptoKeys – *"For distributed TOEs see chapter 2.4.1.1.*

For non-distributed TOEs, the evaluator shall ensure the TSS lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed."

The TSS states in section 8.4.1 that the Admin role is the only role that is permitted to manipulate cryptographic data on the TOE. Cryptographic management functions are performed using the CLI or web GUI commands. Within the TSF, this behavior is limited to generate, import, and delete of X.509 certificates to support TLSC and TLSS mutual authentication, and the import and deletion of SSH public keys for authentication. This assurance activity is considered satisfied as the required information has been discovered.

FMT_SMF.1 – *"The evaluator shall examine the TSS, Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE. The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).*

The evaluator shall examine the TSS and Guidance Documentation to verify they both describe the local administrative interface. The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

For distributed TOEs with the option 'ability to configure the interaction between TOE components' the evaluator shall examine that the ways to configure the interaction between TOE components is detailed in the TSS and Guidance Documentation. The evaluator shall check that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and Guidance Documentation."

Section 8.4.1 of the ST provides a table which identifies which user role that is allowed to perform a particular security management function while indicated which administrative interface (local CLI, SSH CLI, Web GUI) the function can be performed on. Each function has been verified through testing using the instructions provided in the AGD.

This activity passes as the description identifies each security management function, which role has the ability to perform this function, which interface the function can be performed from, and describes the local administrative interface. Additionally, these functions are consistent with Section 6 of the ST, the AGD document which describes all management interfaces, and affirmed through testing.

FMT_SMR.2 – *"The evaluator shall examine the TSS to determine that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE."*

Section 8.4.2 of the ST states the TSF enforces role-based access control (RBAC) to limit access to TSF functions and data based on the set of permissions bound to the subject. The TOE has two administrative roles for the PP defined management functions:

- Administrator – has the ability to perform all PP defined management functions

- Provision – administrative abilities are limited to updating TOE software

This activity passes as the description identifies the supported roles and defines which roles are considered the security administrator and for what functions.

FPT_APW_EXT.1 – *"The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note."*

Section 8.5.1 of the TSS states that no authentication passwords are stored by the TOE in plaintext. All authentication passwords are hashed using SHA-512. There is no function provided by the TOE to display a password value in plaintext nor is the password data recoverable.

This activity passes as the description covers that no passwords are stored in plaintext, the method of obscuring passwords for storage is using a SHA-512 Hash, and no interface is provided to view or recover password data.

FPT_SKP_EXT.1 – *"The evaluator shall examine the TSS to determine that it details how any preshared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured."*

Section 8.5.2 of the TSS states that the TSF prevents unauthorized disclosure of pre-shared keys, symmetric keys and private keys as it does not provide any interface mechanism (CLI or WebGUI) to view these items from volatile memory or file system storage. However, Security Administrators that have the ability to escalate to root privileges, using the sudo command, can have authorized access to the file locations where the secret keys, private keys, and secret key data are stored.

This activity passes as the description identifies that the TOE does not provide specific interface for the purpose of viewing keys in either of the stored locations (RAM or Filesystem).

FPT_STM_EXT.1 – *"The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time, and that it provides a description of how the time is maintained and considered reliable in the context of each of the time related functions."*

Section 8.5.3 of the TSS states that the TOE provides its own time via its internal clock that can be adjusted manually by a Security Administrator via the web GUI. The TOE can also be configured to use an NTP Server as a time source. The TOE uses the clock for several security-relevant purposes, including:

- Audit record timestamps (seconds, milliseconds, microseconds, or nanoseconds).
- X.509v3 certificate validation
- Inactivity of remote sessions
- Inactivity of local session

Section 8.2.8 of the ST states that the TOE uses SHA-384 message digest algorithm to verify the authenticity of the timestamp which ensures reliability.

The activity passes as the description identifies each security function that uses the time and provides a description that time is maintained either manually or using an NTP server which uses SHA-384 message digest algorithm to verify the authenticity of the timestamp to ensure reliability.

FPT_TST_EXT.1 – *"The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.*

For distributed TOEs the evaluator shall examine the TSS to ensure that it details which TOE component performs which self-tests and when these self tests are run."

Section 8.5.4 of the ST describes in detail the TSFs self-tests run at boot. The tests include the standard Linux Filesystem check that verifies RAM and filesystems, a Software Integrity that checks the current state of the constant files on the root partition against the manifest file, that was generated and included in the software as part of the build process, using SHA-384 hashes for comparison, and a Cryptographic Check that performs known answer tests.

These tests are sufficient to validate the correct operation of the TOE because the self-tests are designed to discover any anomalies that would cause the software to be executed in an unpredictable or inconsistent manner. These tests provide assurance that the software has not been tampered with, the filesystem is mounted and validated, and the cryptography is operating correctly.

This activity passes as the details on the cryptographic tests are provided in enough detail and the TSS provides rationale as to why these tests are sufficient to ensure that the TSF is operating correctly.

FPT_TUD_EXT.1 – *"The evaluator shall verify that the TSS describe how to query the currently active version. If a trusted update can be installed on the TOE with a delayed*

activation, the TSS needs to describe how and when the inactive version becomes active. The evaluator shall verify this description.

The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system firmware and software (for simplicity the term 'software' will be used in the following although the requirements apply to firmware and software). The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. Alternatively, an approach using a published hash can be used. In this case the TSS shall detail this mechanism instead of the digital signature verification mechanism. The evaluator shall verify that the TSS describes the method by which the digital signature or published hash is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature or published hash of the update, and the actions that take place for both successful and unsuccessful signature verification or published hash verification.

If the options 'support automatic checking for updates' or 'support automatic updates' are chosen from the selection in FPT_TUD_EXT.1.2, the evaluator shall verify that the TSS explains what actions are involved in automatic checking or automatic updating by the TOE, respectively.

For distributed TOEs, the evaluator shall examine the TSS to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component. Alternatively, this description can be provided in the guidance documentation. In that case the evaluator should examine the guidance documentation instead.

If a published hash is used to protect the trusted update mechanism, then the evaluator shall verify that the trusted update mechanism does involve an active authorization step of the Security Administrator, and that download of the published hash value, hash comparison and update is not a fully automated process involving no active authorization by the Security Administrator. In particular, authentication as Security Administration according to FMT_MOF.1/ManualUpdate needs to be part of the update process when using published hashes."

Section 8.5.5 of the TSS states that the currently executing version of the TOE's software is displayed immediately following successful authentication on all administrative interfaces.

The Security Administrator must download the TOE's update image from the Adtran Customer Portal page to the application server or local workstation. The administrator must use a computer separate from the TOE to recompute the hash of the downloaded image and verify it matches the published hash obtained from the Customer Portal page. Once this validation is complete, the administrator must sign the validated software,

using the end user's approved code signing X.509v3 certificate. This creates the trusted update package.

Once the code signing certificate is imported and marked as trusted, the administrator must fetch the trusted update package from the application server or administrator workstation using the Web GUI. Upon downloading, the TSF will validate the package. If the package validation is successful the trusted update is loaded into the standby area where it will reside dormant until the administrator activates that image (delayed activation). If the validation fails the package is deleted from the TOE.

The currently executing version of the TOE is displayed as well as the version of the image in the standby area. The previous version of the TOE's software is still available for reactivation on the system via the security administrator at any time. Two images remain on the machine until the standby image is either deleted or replaced.

The TOE does not automatically check for software updates for the system.

The activity passes at it fully describes the trusted update process to include the use of a digital signature prior to installation. The description includes the details for a delayed installation and how the current version is displayed automatically.

FPT_TUD_EXT.2 – The evaluator shall verify that the TSS contains a description of how the certificates are contained on the device. The evaluator also ensures that the TSS (or guidance documentation) describes how the certificates are installed/updated/selected, if necessary.

The evaluator shall verify that the TSS describes how the TOE reacts if X.509 certificates are used for trusted updates and the Security Administrator attempts to perform the trusted update using an expired certificate.

The TSS shall describe the point at which revocation checking is performed and describe whether the Security Administrator can manually provide revocation information. It is expected that revocation checking is performed when a certificate is used when performing trusted updates. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device.

Section 8.5.5 of the TSS states that the administrator must import the certificate authority (CA) certificates for the code signing certificate and mark the certificate as trusted.

The TSF validates the package by validating the code signing certificate inside the package using the rules outline in FIA_X509_EXT.1/REV, including the CRL revocation checking, and then verifying the digital signature that was applied to software package. The determination to place the code into the standby area is based on the following:

- If the certificate is deemed invalid (e.g. expired or revoked), the image is not installed and is removed from the system.

- If the certificate is deemed valid, the TSF will then validate the digital signature applied to the code:
    - If the digital signature is not valid, the image is not installed and is removed from the system.
    - If the digital signature check succeeds, the software image is placed in the Standby Area.

The activity passes at it describes how the certificate used to sign the code is stored onto the TOE, how the TOE reacts to the certificate being invalid for any reason including being expired, identifies that the certificate is validated per FIA_X509_EXT.1/REV which outlines that the revocation happens as the last step to the certificate validation.

FTA_SSL_EXT.1 – *"The evaluator shall examine the TSS to determine that it details whether local administrative session locking or termination is supported and the related inactivity time period settings."*

Section 8.6.1 of the ST states that a local CLI session will be automatically terminated due to inactivity, according to the session inactivity timer's value set by the TOE's Security Administrator. The inactivity time period for a local session can be configured between 30 – 3600 seconds.

This activity passes as the description includes a description of the inactivity termination for the local interface along with the inactivity time period range.

FTA_SSL.3 – *"The evaluator shall examine the TSS to determine that it details the administrative remote session termination and the related inactivity time period."*

Section 8.6.2 of the ST states that the TOE will terminate a remote session for both the Remote CLI and Web GUI interfaces due to inactivity according to each interface's respective session inactivity timer configuration. The inactivity time period for a remote session can be configured between 30 – 3600 seconds.

This activity passes as the description includes a description of the inactivity termination for the remote interfaces along with the inactivity time period range.

FTA_SSL.4 – *"The evaluator shall examine the TSS to determine that it details how the local and remote administrative sessions are terminated."*

Section 8.6.3 of the TSS states a Web GUI user may terminate their own sessions by pressing "Logout" under the account button in the top right corner of the screen. The CLI user may terminate their own session by navigating to the menu and selecting "Quit".

This activity passes as all interfaces are described as having a means to terminate one's own session.

FTA_TAB.1 – *"The evaluator shall check the TSS to ensure that it details each administrative method of access (local and remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS). The evaluator shall check the TSS to ensure that all administrative methods of access available to the Security Administrator are listed and that the TSS states that the TOE is displaying an advisory notice and a consent warning message for each administrative method of access. The advisory notice and the consent warning message might be different for different administrative methods of access and might be configured during initial configuration (e.g. via configuration file)."*

Section 8.6.4 of the ST states that there are three possible administrative ways to log into the TOE: locally via physical connection to access the Local CLI, remotely via SSH connection to access the Remote CLI, and remotely using the Web GUI which establishes a HTTPS connection. When logging in locally or remotely, the pre-authentication banner is displayed and must be acknowledged prior to authentication.

This activity passes as each administrative interface is covered and a warning banner is displayed for each of the interfaces.

FTP_ITC.1 – *"The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each secure communication mechanism is identified in terms of the allowed protocols for that IT entity, whether the TOE acts as a server or a client, and the method of assured identification of the non-TSF endpoint. The evaluator shall also confirm that all secure communication mechanisms are described in sufficient detail to allow the evaluator to match them to the cryptographic protocol Security Functional Requirements listed in the ST."*

Section 8.7.1 of the ST states the TOE, acting as the TLS client, uses the TLS protocol to initiate and establish the trusted channel to the Audit Server.

The activity passes as the description states that the TOE acts as a TLS client when communicating with the audit server. This is the only claimed OE connection and the defined protocol is consistent with the claims made in the ST.

FTP_TRP.1/Admin – *"The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST."*

Section 8.7.2 of the TSS states that… Remote administration of the TOE is secured by the utilization of SSH and HTTPS protocols. An HTTPS connection is used for establishing a connection from the Remote Management Workstation to the TOE's Web GUI. An SSH connection is used for establishing a connection from the Remote Management Workstation to the TOE's Remote CLI.

The activity passes as the description states that the TOE acts as a TLS server and SSH server when receiving connections from an administrative workstation for administrative interfaces. These defined protocols are consistent with the claims made in the ST.

2 Operational Guidance Assurance Activities

The evaluation team completed the testing of the Operational Guidance, which includes the review of the *Adran's FSP 3000R7 Supplemental Administrative Guidance* (AGD) document, and confirmed that the Operational Guidance contains all Assurance Activities as specified by the 'Collaborative Protection Profile for Network Devices, version 2.2e (NDcPP)'. The evaluators reviewed the NDcPP 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 NDcPP 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.

Note: Since the TOE is not distributed, AGD Assurance Activities for distributed TOEs have been omitted for clarity.

FAU_GEN.1 – *"The evaluator shall check the guidance documentation and ensure that it provides an example of each auditable event required by FAU_GEN.1 (i.e. at least one instance of each auditable event, comprising the mandatory, optional and selection-based SFR sections as applicable, shall be provided from the actual audit record).*

The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it."

Section 8 of the AGD contains a table of auditable events (Table 4) that is consistent with the auditable events table in the NDcPP for the claimed SFRs. This table includes examples of audit records for different situations that are associated with the requirement including all audit events defined in Table 6-2 of the NDcPP as well as the management actions to configure the TSF capability. Section 8 provides examples of audit records before this table and breaks it down into the individual fields that are prescribed by FAU_GEN.1.2. From this example, the relationship between the audit logs shown in the table and the required fields can be determined clearly.

The AGD was developed with the intent to provide the specific guidance for managing TOE functionality or a pointer to the necessary documentation as defined by the Intended Audience statement in Section 2: "This document is intended for administrators responsible for installing, configuring, and/or operating FSP 3000R7 Network Element. Guidance provided in this document allows the reader to deploy the product in an

environment that is consistent with the configuration that was evaluated as part of the product's Common Criteria (CC) testing process. It also provides the reader with instructions on how to exercise the security functions that were claimed as part of the CC evaluation. The reader is expected to be familiar with the Security Target for FSP 3000R7 Network Element and the general CC terminology that is referenced in it.

This document references the Security Functional Requirements (SFRs) that are defined in the Security Target document and provides instructions on how to perform only the security functions that are defined by these SFRs. Additionally, this document includes references to FSP 3000R7's standard documentation set for the product which contains functionality that is outside the scope of the evaluation. The FSP 3000R7 product, as a whole, provides a great deal of security functionality but only those functions that were in the scope of the claimed PP are discussed here. Any functionality that is not described in this supplemental document or in the FSP 3000R7 Network Element Security Target was not evaluated and should be exercised at the user's risk."

The activity passes as the description states that the AGD provides example audit records for each of the events identified in the AGD. The AGD also provides instructions for the administrative actions related to TSF data related to configuration changes and the necessary TOE mechanisms to enforce the requirements specified in the cPP.

FAU_GEN.2 – *"The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1."*

FAU_STG_EXT.1 – "*The evaluator shall also examine the guidance documentation to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.*

The activity passes as Section 6.1 step 17 of the AGD contains the procedures to establish the secure communications between the TOE and the syslog server. Section 8.1 contains the procedures to configure the TOE to transmit audit records to a specific syslog.

The evaluator shall also examine the guidance documentation to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.

The activity passes as Section 8.1.1 of the AGD states that the TOE then securely transmits audit data via a TLS channel to the external Audit Server in the Operational Environment without administrator intervention. During a connection outage to the Audit Server, the TOE continues to save audit data locally. Once the connection to the Audit

Server is re-established, the TOE automatically starts forwarding new audit records. The TOE does not forward the records created during the outage.

The evaluator shall also ensure that the guidance documentation describes all possible configuration options for FAU_STG_EXT.1.3 and the resulting behaviour of the TOE for each possible configuration. The description of possible configuration options and resulting behaviour shall correspond to those described in the TSS."

This activity passes as Section 8.1 of the AGD details the behavior of the only method for when the TOE performs an audit file rotation. The AGD does provide steps to view audit records on the TOE. There are no configurable options available.

FCS_CKM.1 - *"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 cryptographic protocols defined in the Security Target."*

The activity passes as Section 6.1 of the AGD provides the steps to generate a certificate for the TOE's use. This section also states that **t**he administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_CKM.2 – *"The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s)."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_CKM.4 – "*A TOE may be subject to situations that could prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS (and any other supporting information used). The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.*

For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-levelling and garbage collection. This may result in additional copies of the key that are logically inaccessible but persist physically. Where available, the TOE might then describe use of the TRIM command3 and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance)."

The activity passes as Section 6.1 of the AGD states there is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality. The TOE is not subject to any situations that would prevent or delay key destruction and strictly conforms to the key destruction requirements.

FCS_COP.1/DataEncryption – *"The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected mode(s) and key size(s) defined in the Security Target supported by the TOE for data encryption/decryption."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_COP.1/SigGen – *"The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected cryptographic algorithm and key size defined in the Security Target supported by the TOE for signature services."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_COP.1/Hash - *"The evaluator checks the AGD documents to determine that any configuration that is required to configure the required hash sizes is present."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication. This section also states that the administrator installing

the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_COP.1/KeyedHash – *"The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the values used by the HMAC function: key length, hash function used, block size, and output MAC length used defined in the Security Target supported by the TOE for keyed hash function."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_NTP_EXT.1.1 – "*The evaluator shall examine the guidance documentation to ensure it provides the Security Administrator instructions as how to configure the version of NTP supported, how to configure multiple NTP servers for the TOE's time source and how to configure the TOE to use the method(s) that are selected in the ST."*

The activity passes as Section 7.7.2 of the AGD provides the steps to configure the TOE for NTP usage. The TOE only uses NTPv4 which does not require configuration.

FCS_NTP_EXT.1.2 – *"For each of the secondary selections made in the ST, the evaluator shall examine the guidance document to ensure it instructs the Security Administrator how to configure the TOE to use the algorithms that support the authenticity of the timestamp and/or how to configure the TOE to use the protocols that ensure the integrity of the timestamp."*

The activity passes as Section 7.7.2 of the AGD provides the steps to configure the TOE for NTP Server Authentication.

FCS_NTP_EXT.1.3 – *"The evaluator shall examine the guidance documentation to ensure it provides the Security Administrator instructions as how to configure the TOE to not accept broadcast and multicast NTP packets that would result in the timestamp being updated."*

The activity passes as Section 7.7.2 of the AGD provides the steps to configure the TOE for NTP usage including a NOTE that states the TOE does not accept broadcast and multicast NTP packets.

FCS_RBG_EXT.1 – *"The evaluator shall confirm that the guidance documentation contains appropriate instructions for configuring the RNG functionality."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use the enhanced Security Mode. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_HTTPS_EXT.1 – *"The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to configure TOE for use as an HTTPS client or HTTPS server."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including TLS/HTTPS. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_SSHS_EXT.1.1 – This SFR does not contain any NDcPP AGD Assurance Activities.

FCS_SSHS_EXT.1.2 – This SFR does not contain any NDcPP AGD Assurance Activities.

FCS_SSHS_EXT.1.3 – This SFR does not contain any NDcPP AGD Assurance Activities.

FCS_SSHS_EXT.1.4 – *"The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements)."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including SSH. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the

configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_SSHS_EXT.1.5 - *"The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements)."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including SSH. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

Additionally, Section 7.1 provides the instructions for configuration the TOE to accept public key authentication.

FCS_SSHS_EXT.1.6 – *"The evaluator shall also check the guidance documentation to ensure that it contains instructions to the Security Administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed)."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including SSH. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_SSHS_EXT.1.7 – *"The evaluator shall also check the guidance documentation to ensure that it contains instructions to the Security Administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections with the TOE."*

The activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including SSH. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_SSHS_EXT.1.8 – *"If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, then the evaluator shall check that the guidance documentation describes how to configure those thresholds. Either the allowed values are specified in the guidance documentation and must not exceed the limits specified in the SFR (one hour of session time, one gigabyte of transmitted traffic) or the TOE must not accept values beyond the limits specified in the SFR. The evaluator shall check that the guidance documentation describes that the TOE reacts to the first threshold reached."*

The activity passes as Section 7.1 of the AGD states that the SSH rekey time and size threshold parameters are not administratively configurable. The TSF enforces the connection to be rekeyed after no longer than one hour, and no more than one gigabyte of transmitted data, whichever threshold is reached first.

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

This activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including TLSC (syslog usage). This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_TLSC_EXT.1.2 – *"The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator shall ensure that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.*

Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1, the SFR selects attributes from RFC 5280, and FCO_CPC_EXT.1.2 selects "no channel"; the evaluator shall verify the guidance provides instructions for establishing unique reference identifiers based on RFC5280 attributes."

This activity passes as Section 8.1.1 of the AGD includes steps to set the IP address for the syslog server and states in a NOTE the TOE only supports IPv4 addresses in the Common Name (CN) or the Subject Alternative Name (SAN) extension. Thus, the X.509v3 certificate MUST contain an IPv4 address in the CN or SAN extension utilizing the octet format. The TOE does not support the use of wildcards in the CN or SAN.

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

FCS_TLSC_EXT.1.4 – *"If the TSS indicates that the Supported Elliptic Curves/Supported Groups Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Elliptic Curves/Supported Groups Extension."*

This activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including TLSC (syslog usage). This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_TLSS_EXT.1.1 – *"The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements)."*

This activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including TLSS (web GUI usage). This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_TLSS_EXT.1.2 – *"The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance."*

This activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including TLSS (web GUI usage). This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

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

This activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including TLSS (web GUI usage) and ensuring key curve name is set to secp384r1. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_TLSS_EXT.1.4 – TD0569 "*The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.*

Remark: If multiple contexts are supported for session resumption, the session ID or session ticket may be obtained in one context for resumption in another context. It is possible that one or more contexts may only permit the construction of sessions to be reused in other contexts but not actually permit resumption themselves. For contexts which do not permit resumption, the evaluator is required to verify this behaviour subject to the description provided in the TSS. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session."

This activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including TLSS (web GUI usage) and ensuring Session Resumption is disabled. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2 – *"If the TSS indicates that mutual authentication using X.509v3 certificates is used, the evaluator shall verify that the AGD guidance includes instructions for configuring the client-side certificates for TLS mutual authentication.*

The evaluator shall verify the guidance describes how to configure the TLS client certificate authentication function. If the TSF supports fallback authentication functions, the evaluator shall verify the guidance provides instructions for configuring the fallback authentication functions. If fallback authentication functions can be disabled, the evaluator shall verify the guidance provides instructions for disabling the fallback authentication functions."

This activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including TLSS (web GUI usage) and enabling mutual authentication. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FCS_TLSS_EXT.2.3 – *"The evaluator shall ensure that the AGD guidance describes the configuration of expected identifier(s) for X.509 certificate-based authentication of TLS clients. The evaluator ensures this description includes all types of identifiers described in the TSS and, if claimed, configuration of the TOE to use a directory server."*

This activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including TLSS (web GUI usage) and enabling mutual authentication. This section also states that the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality.

FIA_AFL.1 – *"The evaluator shall examine the guidance documentation to ensure that instructions for configuring the number of successive unsuccessful authentication attempts and time period (if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each "action" specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.*

The evaluator shall examine the guidance documentation to confirm that it describes, and identifies the importance of, any actions that are required in order to ensure that administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA_AFL.1."

The activity passes as Section 7.2 and 6.1 step 4 of the AGD define the steps to configure both the authentication failure handling thresholds and the automatic unlock timer.

FIA_PMG_EXT.1 – *"The evaluator shall examine the guidance documentation to determine that it:*

a) identifies the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords, and

b) provides instructions on setting the minimum password length and describes the valid minimum password lengths supported."

The activity passes as Section 4.1, 7.4, and 4.1 of the AGD provide the instructions to configure the password length and identifies the special characters the TOE supports.

FIA_UAU.7 – *"The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps to ensure authentication data is not revealed while entering for each local login allowed."*

The activity passes as Section 7.4 of the AGD states that password information is never revealed during the authentication process including during login failures.

FIA_UAU_EXT.2 – "*Evaluation Activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1."*

FIA_UIA_EXT.1 – "*The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps (e.g., establishing credential material such as preshared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the guidance documentation provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the guidance documentation provides sufficient instruction on limiting the allowed services."*

This activity passes as Section 6.1 of the AGD provides instructions for the overall establishment and creation of certificates, importing certificates, creating users, Section 7.1.1 has instructions for creating preshared for SSH public key authentication. Section 7.6 has the steps for customizing and enabling the banner. These instruction when fully followed covers all three administrative interfaces.

FIA_X509_EXT.1/Rev – *"The evaluator shall also ensure that the guidance documentation describes where the check of validity of the certificates takes place, describes any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) and describes how certificate revocation checking is performed and on which certificate."*

This activity passes as Section 6.1 of the AGD provides instructions for the overall establishment and creation of certificates, importing certificates. Section 6.3 discusses in detail the validation process X509 certificates used for trusted communication and code signing.

FIA_X509_EXT.2 – *"The evaluator shall also ensure that the guidance documentation describes the configuration required in the operating environment so the TOE can use the certificates. The guidance documentation shall also include any required configuration on the TOE to use the certificates. The guidance document shall also describe the steps for the Security Administrator to follow if the connection cannot be established during the validity check of a certificate used in establishing a trusted channel."*

This activity passes as Section 6.1 of the AGD provides instructions for the overall establishment and creation of certificates, importing certificates. Section 6.3 discusses in detail the validation process X509 certificates used for trusted communication and code signing. Section 8.1 provides the necessary information for the syslog server to create a certificate that will work with the TOE. The AGD does not contain any administrative actions as the Security Administrator does not directly take any action when connection cannot be established during the validity check. The AGD states in Section 6.3 that the TSF does not provide a mechanism to override the validation decision.

FIA_X509_EXT.3 – *"The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certificate Request. If the ST author selects "Common Name", "Organization", "Organizational Unit", or "Country", the evaluator shall ensure that this guidance includes instructions for establishing these fields before creating the Certification Request."*

This activity passes as Section 6.1 of the AGD provides instructions for generating a certificate for the TOE to use.

FMT_MOF.1/ManualUpdate – *"The evaluator shall examine the guidance documentation to determine that any necessary steps to perform manual update are described. The guidance documentation shall also provide warnings regarding functions that may cease to operate during the update (if applicable).*

For distributed TOEs the guidance documentation shall describe all steps how to update all TOE components. This shall contain description of the order in which components need to be updated if the order is relevant to the update process. The guidance documentation shall also provide warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update (if applicable)."

This activity passes as Section 6.2 of the AGD provides steps for validating the software version while Section 7.8.2 provides detailed steps for performing an update. The procedures explain that when package is activated it will automatically cause a reboot.

FMT_MTD.1/CoreData – *"The evaluator shall review the guidance documentation to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the cPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.*

If the TOE supports handling of X.509v3 certificates and provides a trust store, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to configure and maintain the trust store in a secure way. If the TOE supports loading of CA certificates, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to securely load CA certificates into the trust store. The evaluator shall also review the guidance documentation to determine that it explains how to designate a CA certificate a trust anchor."

Section 7.3 of the AGD explains the role-based access control system and that it is enforced on both local and remote authentication. It goes on to state that "All SFR relevant management activity is performed by the Admin, role which corresponds to the NDcPP's definition of Security Administrator. Only users with the Admin role are permitted to create and assign roles to users."

Section 7.1.1 includes the instructions to load the SSH public key for user authentication. Section 6.1 includes instructions for generating public key pair for the TOE to log into the audit server. Section 6.1 includes instructions for importing code signing key and CA certificates for web GUI connection when in mutual authentication mode.

The TSF-data-manipulating functions as required by the PP are contained in FMT_SMF.1. The AGD contained the following:

Management Function	AGD Section
Configure Banner Text	Section 7.6
Configure Idle Session Timeout	Section 7.5.2
Initiate Manual Update	Section 7.8.2
Configure Failed Lockout Threshold	Section 7.2
Configure Lockout Duration	Section 7.2
Manage the cryptographic keys	Section 6.1 Section 7.7.2
Configure the cryptographic functionality	Section 6.1
Re-enable Administrator accounts	Section 7.2
Set time	Section 7.7.1
Configure NTP	Section 6.1 Section 7.7.2
Manage the TOE's trust store and designate X.509v3 certificates as trust anchors	Section 6.1
Ability to manage the trusted public keys database	Section 7.1.1

All functions identified in FMT_SMF.1 have corresponding information on configuring each of the functions. This assurance activity is considered satisfied as the required information has been discovered.

FMT_MTD.1/CryptoKeys – *"For distributed TOEs see chapter 2.4.1.2.*

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed."

The activity passes as Section 7.1.1 includes the instructions to load the SSH public key for user authentication and directions to delete the SSH public key. Section 6.1 includes instructions for generating, importing, and deleting the code signing key and CA certificates for web GUI connection when in mutual authentication mode.

The TOE is a standalone product and therefore the requirements for a distributed TOE are not applicable. This assurance activity is considered satisfied as the required information has been discovered.

FMT_SMF.1 – *"The evaluator shall examine the TSS, Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE. The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).*

The evaluator shall examine the TSS and Guidance Documentation to verify they both describe the local administrative interface. The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

For distributed TOEs with the option 'ability to configure the interaction between TOE components' the evaluator shall examine that the ways to configure the interaction between TOE components is detailed in the TSS and Guidance Documentation. The evaluator shall check that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and Guidance Documentation."

This activity passes as the evaluator was able to create the table below by taking the functions defined the TSS Section 8.4.1 and then mapping the AGD sections to each function.

Management Function	AGD Section
Configure Banner Text	Section 7.6
Configure Idle Session Timeout	Section 7.5.2
Initiate Manual Update	Section 7.8.2
Configure Failed Lockout Threshold	Section 7.2
Configure Lockout Duration	Section 7.2
Manage the cryptographic keys	Section 6.1 Section 7.7.2
Configure the cryptographic functionality	Section 6.1
Re-enable Administrator accounts	Section 7.2

Set time	Section 7.7.1
Configure NTP	Section 6.1 Section 7.7.2
Manage the TOE's trust store and designate X.509v3 certificates as trust anchors	Section 6.1
Ability to manage the trusted public keys database	Section 7.1.1

The evaluator found that the AGD provided instructions for each corresponding functions claimed in the ST. As part of these instructions the AGD provides identification when the administrator must use local administrative interface (for example the initial out-of-the-box setup) or when there is a choice of using CLI (local or SSH) or web GUI. The TOE is a standalone product and therefore the requirements for a distributed TOE are not applicable. The instructions were successfully validated as part of the IND testing effort. This assurance activity is considered satisfied as the required information has been discovered.

FMT_SMR.2 – *"The evaluator shall review the guidance documentation to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration."*

This activity passes as Section 7.1 of the AGD details how to authenticate using local CLI, Remote SSH CLI (password and public key), and web GUI. Section 6.1 provides the steps to configure the TOE to use encrypted communication including TLSS (web GUI usage), SSH (remote CLI).

FPT_APW_EXT.1 – This SFR does not contain any NDcPP AGD Assurance Activities.

FPT_SKP_EXT.1 – This SFR does not contain any NDcPP AGD Assurance Activities.

FPT_STM_EXT.1 – TD0632 – *"The evaluator examines the guidance documentation to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the guidance documentation instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.*

If the TOE supports obtaining time from the underlying VS, the evaluator shall verify the Guidance Documentation specifies any configuration steps necessary. If no configuration is necessary, no statement is necessary in the Guidance Documentation. If there is a delay between updates to the time on the VS and updating the time on the TOE, the evaluator shall ensure the Guidance Documentation informs the administrator of the maximum possible delay."

This activity passes as Section 7.7.1 of the AGD provides instruction on how to set the time manually. Section 7.7.2 provides the steps to configure the use of NTP. The TOE does not obtain time from an underlying VS.

FPT_TST_EXT.1 – *"The evaluator shall also ensure that the guidance documentation describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.*

For distributed TOEs the evaluator shall ensure that the guidance documentation describes how to determine from an error message returned which TOE component has failed the self-test."

The activity passes as Section 6.4 of the AGD explains in detail about the self-test functionality and the possible errors (non-operational state) for example the failure of the cryptographic checks will result in the TOE NOT performing any cryptographic services. This check results in errors identifying the failed cryptographic operations. A failure of this check results in the non-operational state.

FPT_TUD_EXT.1 – *"The evaluator shall verify that the guidance documentation describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the guidance documentation needs to describe how to query the loaded but inactive version.*

This activity passes as Section 7.8.1 of the AGD provides instructions on how to find the currently active version and standby version using all administrative interfaces.

The evaluator shall verify that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification or verification of published hash). The description shall include the procedures for successful and unsuccessful verification. The description shall correspond to the description in the TSS.

This activity passes as Section 7.8.2 of the AGD details the usage of X509 certificate to digitally sign the update. The steps include obtaining the update and published hash from the Adtran Customer Portal, the need to verify the published hash, using an X.509 certificate to apply a digital signature to the update, importing the code signing certificate into the trust store, and fetching the update onto the TOE. The TSF validates the X.509 certificate before verifying the digital signature once the update is on the TOE. If the X.509 certificate is invalid or the digital signature verification fails, the TSF will remove the untrusted package from the system. If the X509 validation and digital signature are valid, the TSF will then place the update into the standby area where it will reside until a Security Administrator activates the update.

If a published hash is used to protect the trusted update mechanism, the evaluator shall verify that the guidance documentation describes how the Security Administrator can obtain authentic published hash values for the updates.

This activity passes as Section 7.8.2 of the AGD states that the update and public hash must be obtained from the Adtran Customer Portal,

For distributed TOEs the evaluator shall verify that the guidance documentation describes how the versions of individual TOE components are determined for FPT_TUD_EXT.1, how all TOE components are updated, and the error conditions that may arise from checking or applying the update (e.g. failure of signature verification, or exceeding available storage space) along with appropriate recovery actions. . The guidance documentation only has to describe the procedures relevant for the Security Administrator; it does not need to give information about the internal communication that takes place when applying updates.

If this was information was not provided in the TSS: For distributed TOEs, the evaluator shall examine the Guidance Documentation to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component.

If this was information was not provided in the TSS: If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the Guidance Documentation contains a description of how the certificates are contained on the device. The evaluator also ensures that the Guidance Documentation describes how the certificates are installed/updated/selected, if necessary."

This activity passes as both section 6.2 and 7.8.2 of the AGD state the code signing certificate must be imported onto the TOE and provide steps to accomplish the import through administrative interfaces.

FPT_TUD_EXT.2 - The evaluator shall verify that the guidance documentation describes how the TOE reacts if X.509 certificates are used for trusted updates and the administrator attempts to perform the trusted update using an expired certificate. The evaluator shall verify any Security Administrator actions related to revocation checking, both accepting or rejecting certificates and manually providing revocation information. The description shall correspond to the description in the TSS.

This activity passes as Section 7.8.2 of the AGD details the usage of X.509 certificate to digitally sign the update and the methodology the TSF uses to validate the X.509 certificate. The determination to place the code into the standby area is based on the following:

- If the certificate is deemed invalid (e.g., expired or revoked), the image is not installed and is removed from the system.
- If the certificate is deemed valid, the TSF will then validate the digital signature applied to the code:
    - If the digital signature is not valid, the image is not installed and is removed from the system.

- If the digital signature check succeeds, the software image is placed in the Standby Area.

If the validation fails, the package is deleted from the TOE. If the validation succeeds, at this point the trusted update has been loaded into the standby area where it will reside dormant until the administrator activates that image (delayed activation); which will result in the reboot of the machine.

FTA_SSL_EXT.1 – *"The evaluator shall confirm that the guidance documentation states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period."*

The assurance activity passes as Section 7.5.2 of the AGD provides instructions on how to set the inactivity timer parameter for local CLI. The TSF will automatically terminate the session when the inactivity threshold is met.

FTA_SSL.3 – *"The evaluator shall confirm that the guidance documentation includes instructions for configuring the inactivity time period for remote administrative session termination."*

The assurance activity passes as Section 7.5.2 of the AGD provides instructions on how to set the inactivity timer parameter for web GUI and remote CLI (same setting as the local cli). The TSF will automatically terminate the session when the inactivity threshold is met.

FTA_SSL.4 – *"The evaluator shall confirm that the guidance documentation states how to terminate a local or remote interactive session."*

The assurance activity passes as Section 7.5.1 of the AGD provides the commands to logout of CLI and web GUI.

FTA_TAB.1 – *"The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message."*

The assurance activity passes as Section 7.6 of the AGD provides the steps to configure the warning banner that is used for all administrative interfaces.

FTP_ITC.1 – *"The evaluator shall confirm that the guidance documentation contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken."*

This activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including TLSC (syslog usage). Section 8.1.1 provides the instructions to enable syslog connection. It is additionally stated that if the connection to the Audit Server is unintentionally broken, no action is required by the administrator to re-establish the connection through the TOE.

FTP_TRP.1/Admin – *"The evaluator shall confirm that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method."*

This activity passes as Section 6.1 of the AGD provides the steps to configure the TOE to use encrypted communication including HTTPS/TLSS. Additionally, Section 7.1 details the step to login remotely using SSH (password or public key) and HTTPS for accessing the web GUI. All remote interfaces are covered.

3 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).

3.1 Platforms Tested and Composition

The evaluation team set up a test environment for the independent functional testing that allowed them to perform all test assurance activities against the SH1HU model over the relevant interfaces. See Section 3.2 Omission Justification for the detailed equivalency analysis to justify omitting the testing of the SH7HU and SH9SU models. Additionally, the following was taken into consideration for scoping the testing:

- **IND Testing: Administrative Interfaces:**

Every administrative interface was used to stimulate the TOE at some point during testing. Not every management function was tested on each interface. The test plan defined which interface was use per test. The interfaces are defined as follows Local CLI (i.e. console connection), Remote CLI, Remote GUI (i.e. Web GUI).

- **IND Testing: Protocol and Functionality:**

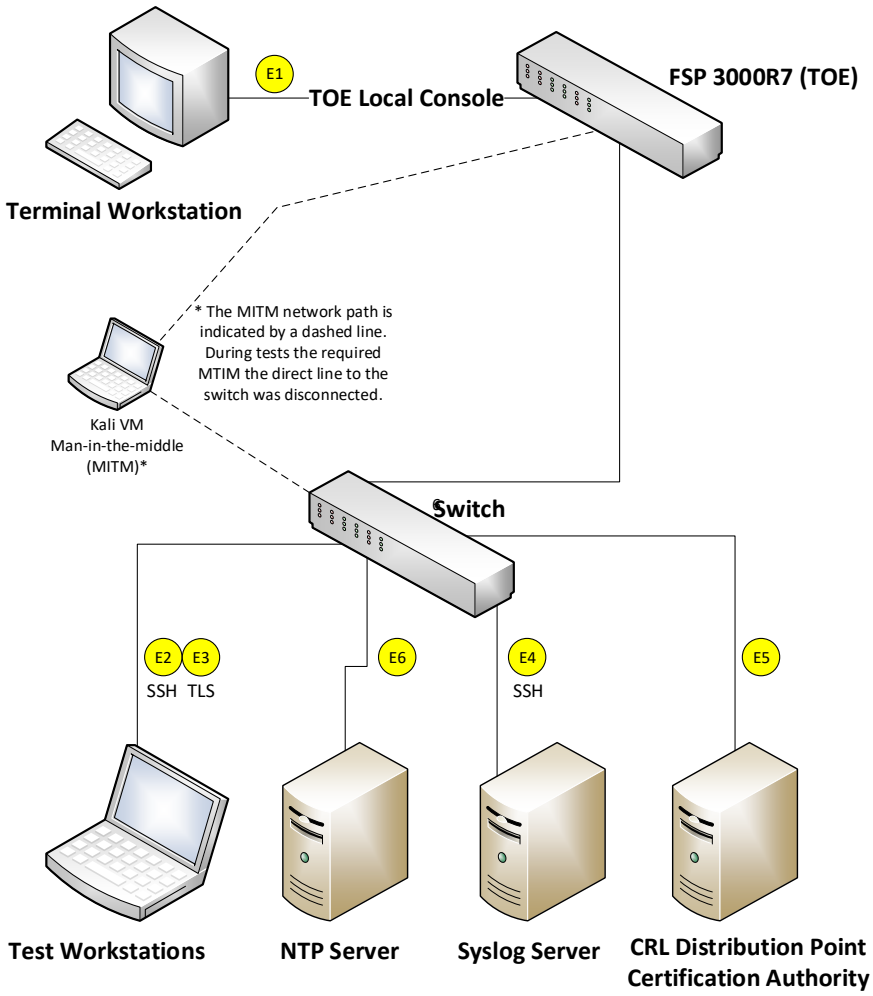
Protocols tested along with the function the protocol supported:

  - TLS Server with and without mutual authentication: Remote Web to TOE (HTTPS/TLSv1.2)
  - TLS Client: TOE to Audit Server (TLSv1.2)
  - SSH Server: Remote CLI to TOE (SSHv2)

- **IND Testing: Regression Testing:**

When the TOE software required updates to fix issues, the evaluation team assessed the appropriate level of regression testing necessary to ensure that any fix did not affect a previously tested functionality. This analysis included impacts to functionality, audit generation, and ST claims. The updates provided by the vendor did not contain new features, but only fixes required for conformance. The "Proprietary_Adtran_NDcPP2.2e_TestMatrix.xlsx" [Test Matrix] contains a running list of the versions tested per test to maintain accurate records.

Test Configuration:



The TOE platforms were configured to communicate with the following environment components:

- Function: Syslog server
  - Platform: ProLiant DL380e Gen8
  - Linux 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64 GNU/Linux
  - Protocols: TLS
  - Interface 4
  - Tools:
    - rsyslogd 8.2102.0 (aka 2021.02)
    - tcpdump 4.99.0

- Function: NTP Server (5)
  - Platform: ProLiant DL380e Gen8
  - OS: Linux 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64 GNU/Linux
  - Protocols: NTPv4
  - Interface 6
  - Tools

  - NTP, NTP2, NTP3: chronyd (chrony) version 4.0 (+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP +SCFILTER +SIGND +ASYNCDNS +NTS +SECHASH +IPV6 -DEBUG)
  - NTP4 : ntpd - NTP daemon program - Ver. 4.2.8p15
  - NTP5: Python 3.9.2 (This host is used to send NTP synch request messages with a spoofed source IP address to a rogue NTP server.)

- Function: OCSP Responder
  - Platform: ProLiant DL380e Gen8
  - OS: Linux 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64 GNU/Linux
  - Protocols: HTTP
  - Interface 5
  - Tools:
    - OpenSSL 1.1.1k
      - OCSP Responder for the tests related to certificate validation.
    - PuTTY SSH Client: version .73 (for rekeying test)
    - tcpdump version 4.99.0

- Function: The following Windows machines were used as the Management Workstations
  - 2 Platforms: HP EliteBook Laptop with
    - Both operating with OS: Windows 10
    - Protocols: TLS, SSH
    - Interfaces 1,2, and 3
    - Tools:
      - WireShark: version 3.6.7
      - Microsoft Edge: version 121.0.2277.112
      - Google Chrome: version 121.0.6167.185
      - PuTTY SSH Client: version .73

- Function: The following Linux machines were used for MTIM, Penetration testing, and Management Workstations
  - 2 Platform: ProLiant DL380e Gen8
    - OS: Linux kali1 4.15.0-kali2-amd64 #1 SMP Debian 4.15.11-1kali1 (2018-03-21)
    - OS: Linux kali 5.18.0-kali5-amd64 #1 SMP PREEMPT_DYNAMIC Debian 5.18.5-1kali6 (2022-07-07) x86_64 GNU/Linux (MITM)
    - x86_64 GNU/Linux
    - Protocols: TLS, SSH
    - Interfaces 2, 3, MITM between switch and TOE
    - Tools:
      - Tcpdump: version 4.99.0
      - OpenSSL version 1.1.1k
      - PuTTY SSH Client: version .73

3.2 Omission Justification

The evaluation team set up a test environment for the independent functional testing that allowed them to perform varying sets of assurance activities against the SH1HU model. Models SH7HU and SH9HU are considered equivalent based on the following analysis:

a. Hardware Assessment:

The boundary of the TOE is the appliance itself. All NDcPP related functionality is contained withing the NCU-3 card where the Network Element Software r22.2.2 is

embedded. All models use the exact same physical management plane hardware (NCU-3) The only difference between the models is the size of the appliance and the number of operational data plane plug-in cards the appliance can hold. The operational plane and respective plug-in cards, have no functions that map to the NDcPP and are therefore out of scope to this evaluation.

Each model used the same processor card: (NXP QorIQ T-Series T1042E)

b. Software Assessment:

Each model uses the exact same binary for installation and the software behaves in the exact same manner on all machines. All the NDcPP defined functionality is contained in the software embedded on the NCU-3 network management plane processor. Each model uses the same cryptographic library: OpenSSL

c. Administrative Interfaces Assessment:

Each model supports the same local console CLI, remote CLI, and web GUI administrative interfaces.

d. Operational Interfaces Assessment:

Each model requires the same operational environment components in order to operate in the evaluated configuration. The TOE's operational environment components include an audit server, NTP server, CA authority to obtain CRLs, a remote administrative workstation, and a terminal for local console connection.

Protocols tested included TLS Server v1.2 with and without mutual authentication enabled, TLS Client, v1.2, and SSH Server v2.

- **Equivalency Conclusion:**

All three models are considered equivalent and therefore only one machine was tested: SH1HU.

3.3 Test Cases

The evaluation team completed the functional testing activities within the laboratory environment. The evaluation team conducted a set of testing that includes all ATE Assurance Activities as specified by the 'collaborative Protection Profile for Network Devices Version 2.2e' (NDcPP) for the SFRs claimed in the Security Target. The evaluators reviewed the NDcPP to identify the security functionality that must be verified through functional testing. This is prescribed by the Assurance Activities for each SFR.

If an SFR is not listed, one of the following conditions applies:

- The Assurance Activity for the SFR specifically indicates that it is simultaneously satisfied by completing a test Assurance Activity for a different SFR.
- The Assurance Activity for the SFR does not specify any actions related to ATE activities (e.g. FPT_APW_EXT.1).

Note that some SFRs do not have Assurance Activities associated with them at the element level (e.g. FPT_TST_EXT.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 AGD. 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.

3.3.1 Security Audit

Test Case Number	001
SFR	FAU_GEN.1
Test Objective	The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the guidance documentation, and that the fields in each audit record have the proper entries. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of auditable events to TOE components in the Security Target. For all events involving more than one TOE component when an audit event is triggered, the evaluator has to check that the event has been audited on both sides (e.g. failure of building up a secure communication channel between the two components). This is not limited to error cases but includes also events about successful actions like successful build up/tear down of a secure communication channel between TOE components. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Create a mapping and verify that all audit records are produced for the various events defined in FAU_GEN.1.1 and Audit Table 16 in the ST.
Test Results	The evaluator observed that all required audit records are generated and the level of information required is present. - Pass
Execution Method	Manual

Test Case Number	002
SFR	FAU_GEN.2
Test Objective	This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1. For distributed TOEs the evaluator shall verify that where auditable events are instigated by another component, the component that records the event associates the event with the identity of the instigator. The evaluator shall perform at least one test on one component where another component instigates an auditable event. The evaluator shall verify that the event is recorded by the component as expected and the event is associated with the instigating component. It is assumed that an event instigated by another component can at least be generated for building up a secure channel between two TOE components. If for some reason (could be e.g. TSS or Guidance Documentation) the evaluator would come to the conclusion that the overall TOE does not generate any events instigated by other components, then this requirement shall be omitted.
Test Instructions	Execute this test per the test steps.
Test Steps	The first part of this test assurance activity is accomplished in conjunction with the testing of FAU_GEN.1.1. The second part of this test assurance activity is not applicable because the TOE is not a distributed TOE.
Test Results	Pass
Execution Method	Manual

Test Case Number	005
SFR	FAU_STG_EXT.1
Test Objective	Testing of the trusted channel mechanism for audit will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall perform the following additional tests for this requirement: a) Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing. The evaluator shall verify that the TOE is capable of transferring audit data to an external audit server automatically without administrator intervention.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Begin capturing packets between the TOE and the remote audit server. 2. Perform some action to cause the TOE to transmit audit data to the remote audit server. 3. Stop capturing packets between the TOE and the remote audit server. 4. Record the name and version of the syslog software used on the audit server. 5. Examine the packet capture and verify that the data transmitted between the TOE and syslog server is protected using TLS. Perform a string search for a field in audit record transmitted (i.e. USER=ADMIN or OPERATION=) within Wireshark and ensure no results.
Test Results	The evaluator observed the audit records from the TOE were received at the syslog server and the traffic was being transferred encrypted between the TOE and the

	syslog server. The evaluator was unsuccessful in finding a character string taken from the audit records received at the syslog server within the captured traffic file as expected. - Pass
Execution Method	Manual

Test Case Number	006
SFR	FAU_STG_EXT.1
Test Objective	Testing of the trusted channel mechanism for audit will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall perform the following additional tests for this requirement: b) Test 2: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.3. Depending on the configuration this means that the evaluator has to check the content of the audit data when the audit data is just filled to the maximum and then verifies that: 1) The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option 'drop new audit data' in FAU_STG_EXT.1.3) 2) The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option 'overwrite previous audit records' in FAU_STG_EXT.1.3) 3) The TOE behaves as specified (for the option 'other action' in FAU_STG_EXT.1.3).
Test Instructions	Execute this test per the test steps.
Test Steps	1. Perform activity to cause the TOE to fill its local audit storage to the maximum capacity. 2. Immediately before the local audit storage data is filled to its maximum capacity, inspect the local audit storage data files by recording their filenames and sizes. 3. Once the local audit storage data is filled to the maximum capacity, observe that the oldest archived log file is deleted. 4. Then observe that the remaining archived log files are rotated. 5. Then observe that the current log file is closed, compressed, and archived. 6. Then observe that a new audit file is opened and receives current log data entries.
Test Results	The evaluator observed that for each log file identified in the ST, the TOE behaved correctly with maintaining the number of archives, deleting oldest audit records (FIFO), and opening a new file upon rollover. - Pass
Execution Method	Manual

Test Case Number	007
SFR	FAU_STG_EXT.1
Test Objective	Testing of the trusted channel mechanism for audit will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall perform the following additional tests for this requirement:

	c) Test 3: If the TOE complies with FAU_STG_EXT.2/LocSpace the evaluator shall verify that the numbers provided by the TOE according to the selection for FAU_STG_EXT.2/LocSpace are correct when performing the tests for FAU_STG_EXT.1.3
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – The TOE does not claim FAU_STG_EXT.2/LocSpace.
Test Results	Pass
Execution Method	Manual

Test Case Number	008
SFR	FAU_STG_EXT.1
Test Objective	Testing of the trusted channel mechanism for audit will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall perform the following additional tests for this requirement: d) Test 4: For distributed TOEs, Test 1 defined above should be applicable to all TOE components that forward audit data to an external audit server. For the local storage according to FAU_STG_EXT.1.2 and FAU_STG_EXT.1.3 the Test 2 specified above shall be applied to all TOE components that store audit data locally. For all TOE components that store audit data locally and comply with FAU_STG_EXT.2/LocSpace Test 3 specified above shall be applied. The evaluator shall verify that the transfer of audit data to an external audit server is implemented.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – The TOE is not a distributed TOE.
Test Results	Pass
Execution Method	Manual

3.3.2 Cryptographic Support

Test cases for FCS_CKM.1 (ECC) , FCS_CKM.2 (ECC), FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, and FCS_RBG_EXT.1 are not included within this section. This is because the ATE Assurance Activities have been satisfied by the vendor having the algorithms in the TOE's cryptographic implementation assessed under the Cryptographic Algorithm Validation Program (CAVP) standard which is governed by a separate validation body than this Common Criteria evaluation. The TOE's CAVP testing directly maps to these SFRs' ATE Assurance Activities. See Cert #A4284 issued August 7, 2023. See table below:

SFR	Algorithm/Protocol	OpenSSL CAVP Cert
FCS_CKM.1	ECC schemes using NIST curves P-384 following FIPS PUB 186-4	ECDSA #A4284
	FFC using safe-prime groups NIST Special Publication 800-56A Revision 3 and RFC 3526.	N/A
FCS_CKM.2	Elliptic curve-based key establishment per NIST Special Publication 800-56A Revision 3	KAS ECC SCC #A4284
	FFC using safe-prime NIST Special Publication 800-56A Revision 3 and groups listed in RFC 3526.	N/A
FCS_COP.1/DataEncryption	AES CTR 256 bits, AES GCM 256 bits	AES#A4284
FCS_COP.1/SigGen	ECDSA FIPS 186-4 Signature Services 384 bits	ECDSA SigGen and SigVer #A4284
FCS_COP.1/Hash	SHA-384 and SHA-512	SHS#A4284

FCS_COP.1/KeyedHash	HMAC-384	HMAC #A4284
FCS_RBG_EXT.1	CTR DRBG (AES-256)	CTR DRBG #A4284

Test Case Number	110
SFR	FCS_CKM.1.1 – TD0580
Test Objective	Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1.
Test Instructions	
Test Steps	**Refer to FCS_CKM.2.1 – Test Case 111**
Test Results	Pass
Execution Method	Manual

Test Case Number	111
SFR	FCS_CKM.2.1 – TD0580
Test Objective	The evaluator shall verify the correctness of the TSF's implementation of safe-prime groups by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses safe-prime groups. This test must be performed for each safe-prime group that each protocol uses.
Test Instructions	
Test Steps	**This test is satisfied by the testing of FTP_TRP.1/Admin, and FCS_SSHS_EXT.1 testing series.**
Test Results	**All FTP_TRP.1/Admin, and FCS_SSHS_EXT.1 testing series were executed successfully and passed. Therefore, this work unit is considered satisfied.**
Execution Method	Manual

Test Case Number	009
SFR	FCS_NTP_EXT.1.1
Test Objective	The version of NTP selected in element 1.1 and specified in the ST shall be verified by observing establishment of a connection to an external NTP server known to be using the specified version(s) of NTP. This may be combined with tests of other aspects of FCS_NTP_EXT.1 as described below.
Test Instructions	Execute this test per the test steps.
Test Steps	The NTP version used by the TOE is confirmed via other test assurance activities in FCS_NTP_EXT.1, as remarked by this test case assurance activity. 1. Ensure the TOE NTP operation mode is set to Disable. 2. Configure the TOE to use NTP v4 and require authentication using SHA384 as the message digest algorithm using the same key configured on the NTP server. 3. Manually set the TOE clock to a value that is different than the clock value set on the NTP server. 4. Begin capturing packets from the TOE. 5. Configure the TOE NTP operation mode to Client. 6. Wait for the TOE to attempt to synchronize its clock with the configured NTP server. 7. Stop capturing packets. 8. Verify that the TOE clock was synchronized to the expected value from the NTP server.

	9. Verify that the TOE used NTP v4 and that it was authenticated.
Test Results	The evaluator observed the TOE connecting to the NTP server using NTPv4 configured to use SHA384 for authenticating. - Pass
Execution Method	Manual

Test Case Number	010
SFR	FCS_NTP_EXT.1.2
Test Objective	The cryptographic algorithms selected in element 1.2 and specified in the ST will have been specified in an FCS_COP SFR and tested in the accompanying Evaluation Activity for that SFR. Likewise, the cryptographic protocol selected in in element 1.2 and specified in the ST will have been specified in an FCS SFR and tested in the accompanying Evaluation Activity for that SFR. [Conditional] If the message digest algorithm is claimed in element 1.2, the evaluator will change the message digest algorithm used by the NTP server in such a way that the new value does not match the configuration on the TOE and confirms that the TOE does not synchronize to this time source. The evaluator shall use a packet sniffer to capture the network traffic between the TOE and the NTP server. The evaluator uses the captured network traffic, to verify the NTP version, to observe time change of the TOE and uses the TOE's audit log to determine that the TOE accepted the NTP server's timestamp update. The captured traffic is also used to verify that the appropriate message digest algorithm was used to authenticate the time source and/or the appropriate protocol was used to ensure integrity of the timestamp that was transmitted in the NTP packets.
Test Instructions	Execute this test per the test steps.
Test Steps	**SHA256 (invalid):** 1. Configure the test NTP server to use SHA256 as the message digest algorithm and a key. 2. Ensure the TOE NTP operation mode is set to Disable. 3. Configure the TOE to use NTP v4 and require authentication using SHA384 as the message digest algorithm and a key. 4. Manually set the TOE clock to a value that is different than the clock value set on the NTP server. 5. Begin capturing packets from the TOE. 6. Configure the TOE NTP operation mode to Client. 7. Wait for the TOE to attempt to synchronize its clock with the configured NTP server. 8. Stop capturing packets. 9. Verify that the TOE clock was NOT synchronized with the NTP server. **SHA384 (valid):** 1. Ensure the TOE NTP operation mode is set to Disable. 2. Configure the TOE to use NTP v4 and require authentication using SHA384 as the message digest algorithm using the same key configured on the NTP server. 3. Manually set the TOE clock to a value that is different than the clock value

	set on the NTP server. 4. Begin capturing packets from the TOE. 5. Configure the TOE NTP operation mode to Client. 6. Wait for the TOE to attempt to synchronize its clock with the configured NTP server. 7. Stop capturing packets. 8. Verify that the TOE clock was synchronized to the expected value from the NTP server. 9. Verify that the TOE used NTP v4 and that it was authenticated.
Test Results	The evaluator confirmed that when the server was configured to use SHA256 as the message digest algorithm the TOE correctly rejected the time change. The evaluator also observed that when the server was configured to use SHA384 as the message digest algorithm, the TOE correctly accepted the time change and updated its time - Pass
Execution Method	Manual

Test Case Number	011
SFR	FCS_NTP_EXT.1.3
Test Objective	The evaluator shall configure NTP server(s) to support periodic time updates to broadcast and multicast addresses. The evaluator shall confirm the TOE is configured to not accept broadcast and multicast NTP packets that would result in the timestamp being updated. The evaluator shall check that the time stamp is not updated after receipt of the broadcast and multicast packets.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Configure the TOE NTP operation mode to Disable. 2. Begin capturing packets from the TOE. 3. Configure the test NTP server to begin transmitting time information using NTP v4 in broadcast server mode. 4. Configure the TOE NTP operation mode to Client. 5. After the TOE receives at least one broadcast NTP packet, stop capturing packets. 6. Verify that the TOE did not update its clock in response to the broadcast NTP packet(s). 7. Configure the TOE NTP operation mode to Disable. 8. Begin capturing packets from the TOE. 9. Configure the test NTP server to transmit time information using NTP v4 in multicast server mode. 10. Configure the TOE NTP operation mode to Client. 11. After the TOE receives at least one multicast NTP packet, stop capturing packets. 12. Verify that the TOE did not update its clock in response to the multicast NTP packet(s).
Test Results	The evaluator observed the TOE correctly fails to synchronize its clock with the NTP server was set to broadcast or multicast modes. - Pass
Execution Method	Manual

Test Case Number	012
SFR	FCS_NTP_EXT.1.4 – TD0528
Test Objective	Test 1: The evaluator shall confirm the TOE supports configuration of at least three

	(3) NTP time sources. The evaluator shall configure at least three NTP servers to support periodic time updates to the TOE. The evaluator shall confirm the TOE is configured to accept NTP packets that would result in the timestamp being updated from each of the NTP servers. The evaluator shall check that the time stamp is updated after receipt of the NTP packets. The purpose of this test to verify that the TOE can be configured to synchronize with multiple NTP servers. It is up to the evaluator to determine that the multi-source update of the time information is appropriate and consistent with the behaviour prescribed by the RFC 1305 for NTPv3 and RFC 5905 for NTPv4.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Configure the TOE NTP client with three distinct NTP servers (i.e., ntp1, ntp2, ntp3) and three distinct keys (i.e., key1, key2, key3) with the mapping of ntp1 to key1, ntp2 to key2, and ntp3 to key3. 2. Begin capturing packets from the TOE. 3. Ensure clock synchronization via NTP is disabled. 4. Ensure that only ntp1 is running. 5. Manually modify the TOE clock to an invalid time value. 6. Enable clock synchronization via NTP. 7. Wait for the clock to synchronize. 8. After the clock has synchronized, repeat Steps 3 through 7, except in Step 3, ensure that only ntp2 is running. 9. After the clock has synchronized, repeat Steps 3 through 7, except in Step 3, ensure that only ntp3 is running. 10. Stop capturing packets.
Test Results	The evaluator observed that the TOE successfully synchronized it's clock with each of the three NTP servers configured. - Pass
Execution Method	Manual

Test Case Number	013
SFR	FCS_NTP_EXT.1.4 – TD0528
Test Objective	Test 2: (The intent of this test is to ensure that the TOE would only accept NTP updates from configured NTP Servers). The evaluator shall confirm that the TOE would not synchronize to other, not explicitly configured time sources by sending an otherwise valid but unsolicited NTP Server responses indicating different time from the TOE's current system time. This rogue time source needs to be configured in a way (e.g. degrade or disable valid and configured NTP servers) that could plausibly result in unsolicited updates becoming a preferred time source if they are not discarded by the TOE. The TOE is not mandated to respond in a detectable way or audit the occurrence of such unsolicited updates. The intent of this test is to ensure that the TOE would only accept NTP updates from configured NTP Servers. It is up to the evaluator to craft and transmit unsolicited updates in a way that would be consistent with the behaviour of a correctly-functioning NTP server.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Ensure the TOE NTP operation mode is set to Enable. 2. Begin capturing packets from the TOE. 3. Begin capturing packets from the rogue NTP server.

	4. Induce the rogue NTP server to send a timestamp update message to the TOE by sending a packet to the rogue NTP server with a forged source IP address matching that of the TOE. 5. After the forged packet and rogue NTP server response timestamp update packets are sent, stop capturing packets. 6. Verify that the TOE clock was NOT modified in response to the rogue NTP server response timestamp update packet.
Test Results	The evaluator observed that the TOE did not synchronize its clock with the rogue NTP server response. - Pass
Execution Method	Manual

Test Case Number	014
SFR	FCS_HTTPS_EXT.1
Test Objective	This test is now performed as part of FIA_X509_EXT.1/Rev testing. Tests are performed in conjunction with the TLS evaluation activities. If the TOE is an HTTPS client or an HTTPS server utilizing X.509 client authentication, then the certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.
Test Instructions	Execute this test per the test steps.
Test Steps	Per test assurance activity, this test is performed as part of other testing.
Test Results	Pass
Execution Method	Manual

Test Case Number	015
SFR	FCS_SSHS_EXT.1.2 **– TD0631**
Test Objective	Test objective: The purpose of these tests is to verify server supports each claimed client authentication method. Test 1: For each supported client public-key authentication algorithm, the evaluator shall configure a remote client to present a public key corresponding to that authentication method (e.g., 2048-bit RSA key when using ssh-rsa public key). The evaluator shall establish sufficient separate SSH connections with an appropriately configured remote non-TOE SSH client to demonstrate the use of all applicable public key algorithms. It is sufficient to observe the successful completion of the SSH Authentication Protocol to satisfy the intent of this test.
Test Instructions	Execute this test per the test steps.
Test Steps	1. On the test machine, configure the SSH client to authenticate using the ecdsa-sha2-nistp384 public key algorithm: `ssh ADMIN@192.168.1.75 -i .\.ssh\id_ecdsa -o "PreferredAuthentications=publickey" -o "PasswordAuthentication=no" -o "PubkeyAuthentication=yes"` 2. Begin capturing packets between the SSH client and the TOE. 3. Connect to the TOE using the SSH client and confirm that the connection was successful. 4. Stop capturing packets.
Test Results	The evaluator observed the successful client authentication using ecdsa-sha2-nistp384 client public key algorithm resulting in the establishment of the SSH

	connection. - Pass
Execution Method	Manual

Test Case Number	016
SFR	FCS_SSHS_EXT.1.2– TD0631
Test Objective	Test objective: The purpose of these tests is to verify server supports each claimed client authentication method. Test 2: The evaluator shall choose one client public key authentication algorithm supported by the TOE. The evaluator shall generate a new client key pair for that supported algorithm without configuring the TOE to recognize the associated public key for authentication. The evaluator shall use an SSH client to attempt to connect to the TOE with the new key pair and demonstrate that authentication fails.
Test Instructions	Execute this test per the test steps.

Test Steps	1. Generate a new SSH ecdsa-sha2-nistp384 keypair on the test machine:
	
	`ssh-keygen -t ecdsa -b 384`
	
	(Save the output of the generated key to a file (e.g., id_ecdsa2))
	
	2. Using the private key from the keypair generated in Step 1, attempt to authenticate to the TOE via the CLI as the Security Administrator using SSH with a valid username:
	
	`ssh ADMIN@192.168.1.75 -i .\.ssh\id_ecdsa2 -o "PreferredAuthentications=publickey" -o "PasswordAuthentication=no" -o "PubkeyAuthentication=yes"`
	
	3. Verify that the authentication attempt to the TOE fails.

Test Results	The evaluator observed that using a new client key pair that has not been installed onto the TOE resulted in a failure to authenticate. - Pass
Execution Method	Manual

Test Case Number	017
SFR	FCS_SSHS_EXT.1.2– TD0631
Test Objective	Test objective: The purpose of these tests is to verify server supports each claimed client authentication method. Test 3: [Conditional] If password-based authentication method has been selected in the FCS_SSHS_EXT.1.2, the evaluator shall configure the TOE to accept password-based authentication and demonstrate that user authentication succeeds when the correct password is provided by the connecting SSH client.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via SSH using a valid username and password. 2. Verify the authentication attempt is successful.
Test Results	The evaluator observed that a user attempting to authenticate to the TOE using a correct password resulted in a successful authentication. - Pass
Execution Method	Manual

Test Case Number	018
SFR	FCS_SSHS_EXT.1.2– TD0631

Test Objective	Test objective: The purpose of these tests is to verify server supports each claimed client authentication method. Test 4: [Conditional] If password-based authentication method has been selected in the FCS_SSHS_EXT.1.2, the evaluator shall configure the TOE to accept password-based authentication and demonstrate that user authentication fails when the incorrect password is provided by the connecting SSH client.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via SSH using a valid username and an invalid password. 2. Verify the authentication attempt is successful.
Test Results	The evaluator observed that a user attempting to authenticate to the TOE using an incorrect password resulted in a failed authentication. - Pass
Execution Method	Manual

Test Case Number	019
SFR	FCS_SSHS_EXT.1.3
Test Objective	The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.
Test Instructions	Execute this test per the test steps.
Test Steps	1. On the test machine, execute the command to send a large packet to the TOE. 2. Verify that the TOE drops any packet larger than the specified size.
Test Results	The evaluator observed when the packet size was larger than the defined number in the ST, the TOE closed the connection. - Pass
Execution Method	Manual

Test Case Number	020
SFR	FCS_SSHS_EXT.1.4
Test Objective	The evaluator must ensure that only claimed ciphers and cryptographic primitives are used to establish an SSH connection. To verify this, the evaluator shall start session establishment for an SSH connection from a remote client (referred to as 'remote endpoint' below). The evaluator shall capture the traffic exchanged between the TOE and the remote endpoint during protocol negotiation (e.g. using a packet capture tool or information provided by the endpoint, respectively). The evaluator shall verify from the captured traffic that the TOE offers all the ciphers defined in the TSS for the TOE for SSH sessions, but no additional ones compared to the definition in the TSS. The evaluator shall perform one successful negotiation of an SSH session to verify that the TOE behaves as expected. It is sufficient to observe the successful negotiation of the session to satisfy the intent of the test. If the evaluator detects that not all ciphers defined in the TSS for SSH are supported by the TOE and/or the TOE supports one or more additional ciphers not defined in the TSS for SSH, the test shall be regarded as failed.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Begin capturing packets between the SSH client test machine and the TOE. 2. Authenticate to the TOE using SSH. 3. Stop capturing packets. 4. Verify an SSH connection was successfully established. 5. Examine the packet capture's "Server: Key Exchange Init" message and verify that no other encryption algorithms other than those claimed in the Security Target are listed in the "encryption_algorithms_server_to_client" string.

Test Results	The ST identifies aes256-gcm@openssh.com as the only encryption algorithm supported. The evaluator observed, from the captured packets, that the aes256-gcm@openssh.com encryption algorithm was the only encryption algorithm used by the TOE when establishing a SSH connection with a non-TOE SSH client. The cipher defined in the ST is consistent with the algorithm that was used for the SSH connection. - Pass
Execution Method	Manual

Test Case Number	021
SFR	FCS_SSHS_EXT.1.5 – TD0631
Test Objective	Test objective: This test case is meant to validate that the TOE server will support host public keys of the claimed algorithm types. Test 1: The evaluator shall configure (only if required by the TOE) the TOE to use each of the claimed host public key algorithms. The evaluator will then use an SSH client to confirm that the client can authenticate the TOE server public key using the claimed algorithm. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. Has effectively been moved to FCS_SSHS_EXT.1.2.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Begin capturing packets between the test machine and the TOE. 2. Authenticate to the TOE via SSH using a ssh client with only ecdsa-sha2-nistp384 selected as the host key algorithm. 3. Stop capturing packets between the test machine and the TOE. 4. Verify that the TOE establishes the SSH connection. 5. Examine packet capture and verify that the ecdsa-sha2-nistp384 public key algorithm was negotiated.
Test Results	The ST identifies ecdsa-sha2-nistp384 as the only host public key algorithm supported. The evaluator observed, from the captured packets, that the ecdsa-sha2-nistp384 host public key algorithm was the only host public key algorithm used by the TOE when establishing a SSH connection with a non-TOE SSH client. The cipher defined in the ST is consistent with the algorithm that was used for the SSH connection. - Pass
Execution Method	Manual

Test Case Number	022
SFR	FCS_SSHS_EXT.1.5 – TD0631
Test Objective	Test objective: This negative test case is meant to validate that the TOE server does not support host public key algorithms that are not claimed. Test 2: The evaluator shall configure a non-TOE SSH client to only allow it to authenticate an SSH server host public key algorithm that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection from the non-TOE SSH client to the TOE SSH server and observe that the connection is rejected.
Test Instructions	Execute this test per the test steps.
Test Steps	1. On the test machine, configure the SSH client use only the ssh-rsa public key algorithm. 2. Begin capturing packets between the SSH client test machine and the TOE. 3. Authenticate to the TOE via the CLI as the Security Administrator using SSH.

	4. Stop capturing packets between the SSH client test machine and the TOE. 5. Verify that the TOE rejects the SSH connection. 6. Examine packet capture and verify that the ssh-rsa encryption algorithm was offered by the test machine (client) in the "server_host_key_algorithms" string.
Test Results	The evaluator observed that when an incorrect host algorithm key was used in an attempt to negotiate a connection, the TOE successfully rejected the connection and generated an audit record with the correct failure reasoning. - Pass
Execution Method	Manual

Test Case Number	023
SFR	FCS_SSHS_EXT.1.6
Test Objective	Test 1: [conditional, if an HMAC or AEAD_AES_*_GCM algorithm is selected in the ST] The evaluator shall establish an SSH connection using each of the algorithms, except "implicit", specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. Note: To ensure the observed algorithm is used, the evaluator shall ensure a non-aes*-gcm@openssh.com encryption algorithm is negotiated while performing this test.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – This conditional test does not apply because HMAC or AEAD_AES_*_GCM is not selected in the Security Target for this SFR.
Test Results	Pass
Execution Method	Manual

Test Case Number	024
SFR	FCS_SSHS_EXT.1.6
Test Objective	Test 2: Test 2: [conditional, if an HMAC or AEAD_AES_*_GCM algorithm is selected in the ST] The evaluator shall configure an SSH client to only allow a MAC algorithm that is not included in the ST selection. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails. Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*-gcm@openssh.com encryption algorithm is negotiated while performing this test.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – This conditional test does not apply because HMAC or AEAD_AES_*_GCM is not selected in the Security Target for this SFR.
Test Results	Pass
Execution Method	Manual

Test Case Number	025
SFR	FCS_SSHS_EXT.1.7
Test Objective	Test 1: The evaluator shall configure an SSH client to only allow the diffiehellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.
Test Instructions	Execute this test per the test steps.
Test Steps	1. On the test machine, configure the SSH client to only use the diffie-hellman-group1-sha1 key exchange algorithm. 2. Begin capturing packets between the SSH client test machine and the

	TOE. 3. Authenticate to the TOE via the CLI as the Security Administrator using the SSH client. 4. Stop capturing packets between the SSH client test machine and the TOE. 5. Using Wireshark, examine the value under the "kex_algorithms" string to verify diffie-hellman-group1-sha1 was offered by the test machine (client). 6. Verify that the SSH connection failed to establish successfully.
Test Results	The evaluator observed that the TOE successfully rejected the connection when diffie-hellman-group1-sha1 key exchange algorithm was used by a non-TOE SSH client in an attempt to establish a connection with the TOE. - Pass
Execution Method	Manual

Test Case Number	026
SFR	FCS_SSHS_EXT.1.7
Test Objective	Test 2: For each allowed key exchange method, the evaluator shall configure an SSH client to only allow that method for key exchange, attempt to connect from the client to the TOE, and observe that the attempt succeeds.
Test Instructions	Execute this test per the test steps.
Test Steps	1. On the test machine, configure the SSH client to only use the diffie-hellman-group15-sha512 key exchange algorithm. 2. Begin capturing packets between the SSH client test machine and the TOE. 3. Authenticate to the TOE via the CLI as the Security Administrator using the SSH client. 4. Stop capturing packets between the SSH client test machine and the TOE. 5. Using Wireshark, examine the value under the "kex_algorithms" string to verify diffie-hellman-groups15-sha512 was used. 6. Verify that the SSH connection established successfully. 7. Expand "SSH Protocol" > "SSH Version 2" > "Key Exchange" > "Algorithms" and examine the value under the "kex_algorithms" string to verify diffie-hellman-groups15-sha512 was used. 8. Repeat Steps 1-6, except in Steps 1 and 6 replace "diffie-hellman-group15-sha512" with "ecdh-sha2-nistp384".
Test Results	The ST identifies diffie-hellman-group15-sha512 and ecdsa-sha2-nistp384 as the only key exchange algorithms supported. The evaluator observed, from the captured packets, that only the diffie-hellman-group15-sha512 and ecdsa-sha2-nistp384 key exchange algorithms were used by the TOE when establishing a SSH connection with a non-TOE SSH client. The ciphers defined in the ST is consistent with the algorithm that was used for the SSH connection. - Pass
Execution Method	Manual

Test Case Number	027
SFR	FCS_SSHS_EXT.1.8
Test Objective	The evaluator needs to perform testing that rekeying is performed according to the description in the TSS. The evaluator shall test both, the time-based threshold and the traffic-based threshold. For testing of the time-based threshold, the evaluator shall use an SSH client to connect to the TOE and keep the session open until the threshold is reached. The evaluator shall verify that the SSH session has been active longer than the threshold value and shall verify that the TOE initiated a rekey (the method of verification

	shall be reported by the evaluator). Testing does not necessarily have to be performed with the threshold configured at the maximum allowed value of one hour of session time, but the value used for testing shall not exceed one hour. The evaluator needs to ensure that the rekeying has been initiated by the TOE and not by the SSH client that is connected to the TOE. For testing of the traffic-based threshold the evaluator shall use the TOE to connect to an SSH client and shall transmit data to and/or receive data from the TOE within the active SSH session until the threshold for data protected by either encryption key is reached. It is acceptable if the rekey occurs before the threshold is reached (e.g. because the traffic is counted according to one of the alternatives given in the Application Note for FCS_SSHS_EXT.1.8). The evaluator shall verify that more data has been transmitted within the SSH session than the threshold allows and shall verify that the TOE initiated a rekey (the method of verification shall be reported by the evaluator). Testing does not necessarily have to be performed with the threshold configured at the maximum allowed value of one gigabyte of transferred traffic, but the value used for testing shall not exceed one gigabyte. The evaluator needs to ensure that the rekeying has been initiated by the TOE and not by the SSH client that is connected to the TOE. If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, the evaluator needs to verify that the threshold(s) can be configured as described in the guidance documentation and the evaluator needs to test that modification of the thresholds is restricted to Security Administrators (as required by FMT_MOF.1/Functions). In cases where data transfer threshold could not be reached due to hardware limitations it is acceptable to omit testing of this (SSH rekeying based on data transfer threshold) threshold if both the following conditions are met: a) An argument is present in the TSS section describing this hardware based limitation and b) All hardware components that are the basis of such argument are definitively identified in the ST. For example, if specific Ethernet Controller or WiFi radio chip is the root cause of such limitation, these chips must be identified.
Test Instructions	Execute this test per the test steps.
Test Steps	**a) Time-based Rekey (1 hour):** 1. Authenticate to the TOE via the CLI as the Security Administrator using SSH with the following command to ensure that the test SSH client does not perform a rekey before the TOE: `ssh -vvv -E ./ssh_client_log ADMIN@[TOE_IP_ADDRESS] -o "RekeyLimit=10G 10h"` 2. Wait 1 hour and verify that the TOE generates an audit record for the SSH rekey performed by the TOE.

	b) Traffic-based Rekey (990 MB): 1. Transfer a 990 MB file to the TOE via SSH (i.e. using SCP) with the following command to ensure that the test SSH client does not perform a rekey before the TOE: `scp -vvv -o "RekeyLimit=10G 10h" 990mbfile ADMIN@[TOE_IP_ADDRESS]:/tmp` 2. Verify that the TOE generates an audit record for the SSH rekey performed by the TOE.
Test Results	'The evaluator observed that a timed rekeying event (SSH-CHANNEL-REKEY-INTERVAL) happened in 50 minutes which is less than an hour. The evaluator observed that the data rekeying event (SSH-CHANNEL-REKEY) happened after 944,584,604 bytes were sent which is less than 1 GBytes. - Pass
Execution Method	Manual

Test Case Number	028
SFR	FCS_TLSC_EXT.1.1
Test Objective	Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites 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 HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Instructions	Execute this test per the test steps.
Test Steps	1. Configure the remote server such that only the following ciphersuite is supported: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 2. Begin capturing packets between the TOE and the remote server. 3. Cause the TOE to establish a TLS connection to the remote server. 4. Stop capturing packets between the TOE and the remote server. 5. Inspect the packet capture and verify that the Client Hello message contains the ciphersuite selected in Step 1.
Test Results	The evaluator observed the claimed ciphersuite in the ST was successfully used to connect to the TOE and audits the event. - Pass
Execution Method	Manual

Test Case Number	029
SFR	FCS_TLSC_EXT.1.1
Test Objective	Test 2: The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field, and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.

Test Instructions	Execute this test per the test steps.
Test Steps	1. On the remote server, load the certificate containing the Server Authentication purpose. 2. Begin capturing packets between the TOE and the remote server. 3. Cause the TOE to establish a TLS connection to the remote server. 4. Stop capturing packets between the TOE and the remote server. 5. Inspect the packet capture and verify that the TOE successfully established a connection to the remote server. 6. On the remote server, load the certificate without the Server Authentication purpose. 7. Repeat Steps 2-4. 8. Inspect the packet capture and verify that the TOE failed to establish a connection to the remote server.
Test Results	The evaluator observed that when the correct server authentication purpose is specified in the extendedKeyUsage field the connection is successful. The evaluator also observed that when an the extendedKeyUsage field was missing the TOE correctly terminates the connection. - Pass
Execution Method	Manual

Test Case Number	030
SFR	FCS_TLSC_EXT.1.1
Test Objective	Test 3: The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send an ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite). The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.
Test Instructions	Execute this test per the test steps.
Test Steps	1. On a remote server, load the certificate generated in setup. 2. Run the MITM tool to modify traffic. 3. Begin capturing packets between the TOE and the remote server. 4. Establish a TLS connection between the TOE and the remote server. 5. Stop capturing packets. 6. Inspect the packet capture to verify that a TLS connection could not be established, and that the TOE client disconnected after receiving the server's Certificate handshake message.
Test Results	The evaluator observed that when the remote server sends a RSA server certificate while using the TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ciphersuite the TOE disconnects after receiving the server's Certificate handshake message. - Pass
Execution Method	Manual

Test Case Number	031
SFR	FCS_TLSC_EXT.1.1
Test Objective	Test 4: The evaluator shall perform the following 'negative tests': a) The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection. b) Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after

	receiving the Server Hello. c) [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension the evaluator shall configure the server to perform an ECDHE or DHE key exchange in the TLS connection using a non-supported curve/group (for example P-192) and shall verify that the TOE disconnects after receiving the server's Key Exchange handshake message.
Test Instructions	Execute this test per the test steps.
Test Steps	a) 1. Run the MITM tool to modify traffic. 2. Begin capturing packets between the TOE and the remote server. 3. Establish a TLS connection between the TOE and the remote server. 4. Stop capturing packets. **5.** Verify that the TLS connection could not be established, and the client refused the server's ciphersuite selection. b) 1. Repeat Steps 1-5 in part (a), except using the MITM tool for part (b). c) 1. Repeat Steps 1-5 in part (a), except using the MITM tool for part (c).
Test Results	The evaluator observed that when the remote server selects the TLS_NULL_WITH_NULL_NULL ciphersuite, the TOE correctly terminates the connection. Additionally, when the remote server selects a ciphersuite not presented by the TOE Client Hello message, the TOE correctly terminates the connection after receiving the Server Hello. - Pass
Execution Method	Manual

Test Case Number	032
SFR	FCS_TLSC_EXT.1.1
Test Objective	Test 5: The evaluator performs the following modifications to the traffic: a) Change the TLS version selected by the server in the Server Hello to a non-supported TLS version and verify that the client rejects the connection. b) [conditional]: If using DHE or ECDH, modify the signature block in the Server's Key Exchange handshake message, and verify that the handshake does not finished successfully, 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 Instructions	Execute this test per the test steps.
Test Steps	a) 1. Begin capturing packets between the TOE and the TLS server. 2. Run the MITM tool to modify traffic. 3. Initiate a connection from the TOE to the server. 4. Stop capturing packets. **5.** Confirm the TOE rejects the connection. b) 6. Repeat Steps 1-5 in part (a), except using the MITM tool for part (b).
Test Results	The evaluator observed that when the TLS version selected by the server in the Server Hello was set to a non-supported TLS version, the TOE correctly rejected the connection. Additionally, when the signature block was modified, the TOE

	correctly rejected the connection. - Pass
Execution Method	Manual

Test Case Number	033
SFR	FCS_TLSC_EXT.1.1
Test Objective	Test 6: The evaluator performs the following 'scrambled message tests': a) Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application data flows. b) Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake does not finish successfully and no application data flows. c) Modify at least one byte in the server's nonce in the Server Hello handshake message and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.
Test Instructions	Execute this test per the test steps.
Test Steps	a) 1. Begin capturing packets between the TOE and the TLS server. 2. Run the MITM tool to modify traffic. 3. Initiate a connection from the TOE to the server. 4. Stop capturing packets. 5. Confirm the TOE rejects the connection. b) 6. Repeat Steps 1-5 in part (a), except using the MITM tool for part (b). c) 7. Repeat Steps 1-5 in part (a), except using the MITM tool for part (c).
Test Results	The evaluator observed the TOE correctly terminated the connection when the following happened: a) A byte in the Server Finished handshake message is modified, the handshake does not finish successfully, and no application data flows. b) A garbled message is sent from the server after the server has issued the ChangeCipherSpec message, the handshake does not finish successfully, and no application data flows. c) One byte in the server's nonce in the Server Hello handshake message is modified, and the server denies the client's Finished handshake message. - Pass
Execution Method	Manual

Test Case Number	034
SFR	FCS_TLSC_EXT.1.2
Test Objective	Note that the following tests are marked conditional and are applicable under the following conditions: a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable. or b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable or

	c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable. Note that for some tests additional conditions apply. IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules: • IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986. • IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested. The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: a) Test 1 [conditional]: 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. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN. Remark: 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.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Install a certificate on the server that contains a Common Name (CN) that does not match the reference identifier of the remote server and does not contain the SAN extension. 2. Begin capturing packets between the TOE and the server. 3. Connect the TOE to the server using TLS. 4. Stop capturing packets. 5. Verify that the connection fails.
Test Results	The evaluator observed that when a certificate contained a CN that does not match the reference identifier and no SAN is claimed, the TOE correctly terminated the connection. - Pass
Execution Method	Manual

Test Case Number	035
SFR	FCS_TLSC_EXT.1.2
Test Objective	Note that the following tests are marked conditional and are applicable under the following conditions: a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable. or b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable

	or c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable. Note that for some tests additional conditions apply. IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules: • IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986. • IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested. The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: b) Test 2 [conditional]: 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 (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Install a certificate on the server 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 of the server. 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.
Test Results	The evaluator observed that when a certificate contained a CN that matched the reference identifier and a SAN that did not match the reference identifier, the TOE correctly terminated the session. - Pass
Execution Method	Manual

Test Case Number	036
SFR	FCS_TLSC_EXT.1.2
Test Objective	Note that the following tests are marked conditional and are applicable under the following conditions: a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable. or b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable or c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is

	selected, only test 7 is applicable. Note that for some tests additional conditions apply. IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules: • IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986. • IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested. The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: c) 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. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Install a certificate on the server that contains a CN that matches the reference identifier of the server but does not contain the SAN extension. 2. Begin capturing packets between the TOE and the server. 3. Connect the TOE to the server. 4. Stop capturing packets. 5. Verify the connection succeeds.
Test Results	The evaluator observed that when a certificate used a CN that did match the reference identifier and no SAN declared, the TOE correctly established the connection. - Pass
Execution Method	Manual

Test Case Number	037
SFR	FCS_TLSC_EXT.1.2
Test Objective	Note that the following tests are marked conditional and are applicable under the following conditions: a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable. or b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable or c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable. Note that for some tests additional conditions apply. IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the

	evaluator shall follow the following formatting rules: • IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986. • IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested. The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: d) Test 4 [conditional]: 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. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).
Test Instructions	Execute this test per the test steps.
Test Steps	1. Install a certificate on the server with a CN that does not match the reference identifier but does contain an identifier of the server in the SAN that matches. 2. Begin capturing packets between the TOE and the server. 3. Connect the TOE to the server. 4. Stop capturing packets. 5. Verify the connection succeeds.
Test Results	The evaluator observed that when a certificate used a CN that did not match the reference identifier along with a SAN that did match the reference identifier, the TOE correctly established the connection. - Pass
Execution Method	Manual

Test Case Number	038
SFR	FCS_TLSC_EXT.1.2
Test Objective	Note that the following tests are marked conditional and are applicable under the following conditions: a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable. or b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable or d) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable. Note that for some tests additional conditions apply. IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules: • IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.

	• IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested. The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: e) Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URIID): 1) [conditional]: 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. 2) [conditional]: 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 succeeds, if wildcards are supported, or fails if wildcards are not supported. 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.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)
Test Instructions	Execute this test per the test steps.
Test Steps	**N/A – Per the Security Target, wildcards and DNS entries are not supported in CN or SAN.**
Test Results	Pass
Execution Method	Manual

Test Case Number	039
SFR	FCS_TLSC_EXT.1.2 – TD0790
Test Objective	Note that the following tests are marked conditional and are applicable under the following conditions: a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable. or b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable or c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable. Note that for some tests additional conditions apply. IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules: • IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range

	from 0-255 separated by periods as specified in RFC 3986. • IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested. The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: f) Test 6 [conditional]: If IP address identifiers are supported in the SAN or CN, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with a wildcard asterisk (*) (e.g. CN=*.168.0.1 when connecting to 192.168.0.1,CN=2001:0DB8:0000:0000:0008:0800:200C:* when connecting to 2001:0DB8:0000:0000:0008:0800:200C:417A). The certificate shall not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported IP address version (e.g. IPv4, IPv6). Remark: 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 6.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Using the certificate created during the setup, configure the remote server to present it in response to connection requests. 2. Cause the TOE to initiate a TLS connection to the remote server. 3. Verify the TLS connection between the TOE and the remote server is unsuccessful.
Test Results	The evaluator observed that when a certificate using a wildcard as part of the CN and no SAN, the TOE correctly terminated the connection. - Pass
Execution Method	Manual

Test Case Number	040
SFR	FCS_TLSC_EXT.1.2
Test Objective	Note that the following tests are marked conditional and are applicable under the following conditions: a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable. or b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable or c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable. Note that for some tests additional conditions apply. IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules: • IPv4: The CN contains a single address that is represented a 32-bit

	numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986. • IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested. The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: g) Test 7 [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator modifies each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier): 1) The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails. 2) The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-atserialNumber=correct_identifier, the certificate could instead include id-at-name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: 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 this test. 3) The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. 4) The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)
Test Instructions	Execute this test per the test steps.
Test Steps	**N/A – The Security Target does not claim FPT_ITT.1; therefore, this conditional test, Test 7, does not apply per the test instructions.**
Test Results	Pass
Execution Method	Manual

Test Case Number	041
SFR	FCS_TLSC_EXT.1.3
Test Objective	The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows: Test 1: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.

Test Instructions	Execute this test per the test steps.
Test Steps	1. Using the certificate imported during the setup, configure the remote server to present the full valid certificate. 2. Begin capturing packets between the server and the TOE. 3. Initiate a connection from the TOE to the server. 4. Stop capturing packets between the server and the TOE. 5. Verify connection succeeds.
Test Results	The evaluator observed that TOE the connection was successfully established when a complete certificate chain was presented and the CA certificate was properly imported onto the TOE. - Pass
Execution Method	Manual

Test Case Number	042
SFR	FCS_TLSC_EXT.1.3
Test Objective	The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows: Test 2: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Configure the remote server to present the certificate with one of the intermediates removed from the chain. 2. Begin capturing packets between the server and the TOE. 3. Initiate a connection from the TOE to the server. 4. Stop capturing packets between the server and the TOE. 5. Verify connection fails. The ST selects “Not implement any administrator override mechanism” for FCS_TLSS_EXT.2.2. As such, there are no “selected types of failure defined in the SFR”.
Test Results	The evaluator observed that when the incomplete certificate chain was presented, the TOE terminated the connection and audited the event. - Pass
Execution Method	Manual

Test Case Number	043
SFR	FCS_TLSC_EXT.1.3
Test Objective	The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows: Test 3 [conditional]: The purpose of this test to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative

	override available to accept such certificate.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – This conditional test does not apply as the ST states the TSF shall not implement any administrator override mechanism.
Test Results	Pass
Execution Method	Manual

Test Case Number	044
SFR	FCS_TLSC_EXT.1.4
Test Objective	Test 1 [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) 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.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Configure the remote test server to use the secp384r1 elliptic curve. 2. Begin capturing packets between the TOE and the remote server. 3. Perform some action on the TOE that causes it to initiate a connection to the remote server. 4. Stop capturing packets between the TOE and the remote server. 5. Verify that the TOE accepts the connection.
Test Results	The evaluator observed the TOE connected to the server using the one curve claimed in the ST. - Pass
Execution Method	Manual

Test Case Number	045
SFR	FCS_TLSS_EXT.1.1
Test Objective	Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites 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 HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Instructions	Execute this test per the test steps.

Test Steps	1. Begin capturing packets between the TOE and the test machine.
	2. Execute the following commands on the test machine in order to initiate a TLS connection to the TOE using the specific ciphersuite:
	
	`openssl s_client -connect [TOE_IP_ADDRESS]:443 -cert <client-certificate> -key <client-key> -CAfile <intermediate-CA-chain> -cipher ECDHE-ECDSA-AES256-GCM-SHA384`
	
	`GET / HTTP/1.1`
	`Host: [TOE_IP_ADDRESS]`
	
	3. Stop capturing packets.
	4. Verify the connection succeeded (exchange of Application Data).

Test Results	The evaluator observed the claimed ciphersuite in the ST was successfully used to connect to the TOE. - Pass

Execution Method	Manual

Test Case Number	046
SFR	FCS_TLSS_EXT.1.1
Test Objective	Test 2: The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites 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 ciphersuite and verify that the server denies the connection.
Test Instructions	Execute this test per the test steps.
Test Steps	(a) Unsupported ciphersuites: 1. Begin capturing packets between the TOE and the test machine. 2. Configure the TLS client to use the following list of ciphersuites: `openssl s_client -connect <TOE_IP_ADDRESS>:443 -cert <client-certificate> -key <client-key> -CAfile <intermediate-CA-chain> -cipher RC4-MD5,DES-CBC3-SHA` 3. Initiate a connection from the test machine to the TOE. 4. Stop capturing packets. 5. Verify that the TLS connection could not be established. (b) TLS_NULL_WITH_NULL_NULL: 6. Begin capturing packets between the TOE and the test machine. 7. Run the MITM tool to modify traffic. 8. Initiate a connection from the test machine to the TOE. 9. Stop capturing packets. 10. Verify that the TLS connection could not be established and the server refused to negotiate a ciphersuite.
Test Results	The evaluator observed that when the client offers ciphersuites not claimed in the ST the TOE successfully terminates the connection and audits the event. The evaluator also observed that when the TOE receives a TLS_NULL_WITH_NULL_NULL ciphersuite the TOE successfully terminates the connection and audits the event. - Pass
Execution Method	Manual

Test Case Number	047
SFR	FCS_TLSS_EXT.1.1
Test Objective	Test 3: The evaluator shall perform the following modifications to the traffic: a) Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data. b) (Test Intent: The intent of this test is to ensure that the server's TLS implementation immediately makes use of the key exchange and authentication algorithms to: a) Correctly encrypt (D)TLS Finished message and b) Encrypt every (D)TLS message after session keys are negotiated.)

	The evaluator shall use one of the claimed ciphersuites to complete a successful handshake and observe transmission of properly encrypted application data. The evaluator shall verify that no Alert with alert level Fatal (2) messages were sent. The evaluator shall verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator shall examine the Finished message (encrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 11 22 33 44 55...) and confirm that it does not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c...), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. There is a chance that an encrypted Finished message contains a hexadecimal value of '14' at the position where a plaintext Finished message would contain the message type code '14'. If the observed Finished message contains a hexadecimal value of '14' at the position where the plaintext Finished message would contain the message type code, the test shall be repeated three times in total. In case the value of '14' can be observed in all three tests it can be assumed that the Finished message has indeed been sent in plaintext and the test has to be regarded as 'failed'. Otherwise it has to be assumed that the observation of the value '14' has been due to chance and that the Finished message has indeed been sent encrypted. In that latter case the test shall be regarded as 'passed'.
Test Instructions	Execute this test per the test steps.
Test Steps	**a)** 1. Begin capturing packets between the TOE and the TLS client. 2. Run the MITM tool to modify traffic. 3. Initiate a connection from the TLS client to the TOE. 4. Stop capturing packets. 5. Confirm the TLS connection failed to establish. **b)** 6. Open Wireshark and begin capturing packets between the TOE and the TLS client. 7. Initiate a connection from the TLS client to the TOE. 8. Stop capturing packets. 9. Inspect the packet capture for each of the following: a. Verify the Finished message (Encrypted Handshake) is sent immediately after the server's ChangeCipherSpec message. b. Examine the Finished message and confirm it does not contain unencrypted data (by verifying that the first byte of the Finished message does not equal hexadecimal 14.
Test Results	The evaluator observed that when the handshake message was modified the TOE successfully terminated the connection and audited the event. The evaluator also observed that when the handshake message was not modified that the connection was successful and no data was sent in the clear. The finished message (encrypted handshake) message was immediately after the ChangeCipherSpec and was encrypted. - Pass
Execution Method	Manual

Test Case Number	048

SFR	FCS_TLSS_EXT.1.2
Test Objective	The evaluator shall send a Client Hello requesting a connection for all mandatory and selected protocol versions in the SFR (e.g. by enumeration of protocol versions in a test client) and verify that the server denies the connection for each attempt.
Test Instructions	Execute this test per the test steps.

Test Steps	1. Begin capturing packets between the TOE and the TLS client.
	2. Execute the following commands on the test machine to initiate a connection to the TOE using the disallowed protocols:
	
	`openssl s_client -connect <TOE_IP_ADDRESS>:443 -cert <client-certificate> -key <client-key> -CAfile <intermediate-CA-chain> -ssl2`
	
	`openssl s_client -connect <TOE_IP_ADDRESS>:443 -cert <client-certificate> -key <client-key> -CAfile <intermediate-CA-chain> -ssl3`
	
	`openssl s_client -connect <TOE_IP_ADDRESS>:443 -cert <client-certificate> -key <client-key> -CAfile <intermediate-CA-chain> -tls1`
	
	`openssl s_client -connect <TOE_IP_ADDRESS>:443 -cert <client-certificate> -key <client-key> -CAfile <intermediate-CA-chain> -tls1_1`
	
	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 observed that when unsupported versions of TLS (or SSL) were used the TOE successfully terminated the session. - Pass
Execution Method	Manual

Test Case Number	049
SFR	FCS_TLSS_EXT.1.3
Test Objective	Test 1: [conditional] If ECDHE ciphersuites are supported: a) The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single supported elliptic curve specified in the Elliptic Curves Extension. The Evaluator shall verify (though a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange message and successfully establishes the connection. b) The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g. secp192r1 (0x13)) specified in RFC4492, chap. 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.
Test Instructions	Execute this test per the test steps.
Test Steps	**a)** 1. Load a server certificate onto the TOE that accepts the secp384r1 curve. 2. Begin capturing packets between the TLS client and the TOE. 3. Initiate a connection from the TLS client to the TOE such that the

	supported curve is specified (secp384r1):
	
	`openssl s_client -connect <TOE_IP_ADDRESS>:443 -cert <client-certificate> -key <client-key> -CAfile <intermediate-CA-chain> -curves secp384r1`
	
	4. Stop capturing packets with Wireshark.
	5. Confirm the TOE selects the secp384r1 curve in the Server Key Exchange message and that the connection is successfully established.
	b)
	1. Begin capturing packets between the TLS client and the TOE.
	2. Initiate a connection from the TLS client to the TOE such that the unsupported curve is specified (secp192k1):
	
	`openssl s_client -connect <TOE_IP_ADDRESS>:443 -cert <client-certificate> -key <client-key> -CAfile <intermediate-CA-chain> -curves secp192k1`
	
	3. Stop capturing packets with Wireshark.
	4. Confirm the TOE does not send a Sever Hello message and the connection is not successfully established.

Test Results	The evaluator observed that when the client uses the claimed supported curve the connection was successful. The evaluator also observed that when the client uses a non-claimed supported curve the TOE successfully terminated the connection and audited the event. - Pass
Execution Method	Manual

Test Case Number	050
SFR	FCS_TLSS_EXT.1.3
Test Objective	Test 2: [conditional] If DHE ciphersuites are supported, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use a supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the message are the ones configured Diffie-Hellman parameter size(s).
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – The TOE does not support DHE ciphersuites.
Test Results	Pass
Execution Method	Manual

Test Case Number	051
SFR	FCS_TLSS_EXT.1.3
Test Objective	Test 3: [conditional] If RSA key establishment ciphersuites are supported, the evaluator shall repeat this test for each RSA key establishment key size. If any configuration is necessary, the evaluator shall configure the TOE to perform RSA key establishment using a supported key size (e.g. by loading a certificate with the appropriate key size). The evaluator shall attempt a connection using a supported RSA key establishment ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a certificate whose modulus is

	consistent with the configured RSA key size.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – Per the ST, RSA key establishment ciphersuites are not supported.
Test Results	Pass
Execution Method	Manual

Test Case Number	052
SFR	FCS_TLSS_EXT.1.4 – TD0569
Test Objective	Test Objective: To 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). Remark: If multiple contexts are supported for session resumption, the session ID or session ticket may be obtained in one context for resumption in another context. It is possible that one or more contexts may only permit the construction of sessions to be reused in other contexts but not actually permit resumption themselves. For contexts which do not permit resumption, the evaluator is required to verify this behaviour subject to the description provided in the TSS. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session. Test 1 [conditional]: If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test: a) The client sends a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket. b) The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake). c) The client verifies the Server Hello message contains a zero-length session identifier or passes the following steps: Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID. d) The client completes the TLS handshake and captures the SessionID from the ServerHello. e) e) The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d). f) f) The client verifies the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Begin capturing packets between the TOE and the test machine. 2. Initiate a connection to the TOE by sending a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket: `openssl s_client -connect <TOE_IP_ADDRESS>:443 -cert <client-certificate> -key <client-key> -CAfile <intermediate-CA-chain>`

	3. Stop capturing packets between the TOE and the test machine. 4. Confirm that the TOE does not send a NewSessionTicket handshake message (at any point in the handshake). 5. Confirm that the Server Hello message contains a zero length session identifier.
Test Results	The evaluator observed that the Server Hello responded with a Session ID length of 0. This would be consistent with the claim that the TOE does not support session resumption based on session IDs or session tickets. - Pass
Execution Method	Manual

Test Case Number	053
SFR	FCS_TLSS_EXT.1.4 – TD0569
Test Objective	Test Objective: To 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). Remark: If multiple contexts are supported for session resumption, the session ID or session ticket may be obtained in one context for resumption in another context. It is possible that one or more contexts may only permit the construction of sessions to be reused in other contexts but not actually permit resumption themselves. For contexts which do not permit resumption, the evaluator is required to verify this behaviour subject to the description provided in the TSS. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session. Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): a) The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246). b) The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – The Security Target specifies that the TOE does not support session resumption using session IDs; therefore, this conditional test does not apply.
Test Results	Pass

Execution Method	Manual

Test Case Number	054
SFR	FCS_TLSS_EXT.1.4 – TD0556 & TD0569
Test Objective	Test Objective: To 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). Remark: If multiple contexts are supported for session resumption, the session ID or session ticket may be obtained in one context for resumption in another context. It is possible that one or more contexts may only permit the construction of sessions to be reused in other contexts but not actually permit resumption themselves. For contexts which do not permit resumption, the evaluator is required to verify this behaviour subject to the description provided in the TSS. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session. Test 3 [conditional]: If the TOE supports session tickets according to RFC5077, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): a) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with an abbreviated handshake described in section 3.1 of RFC 5077 and illustrated with an example in figure 2. Of particular note: if the server successfully verifies the client's ticket, then it may renew the ticket by including a NewSessionTicket handshake message after the ServerHello in the abbreviated handshake (which is shown in figure 2). This is not required, however as further clarified in section 3.3 of RFC 5077. b) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – The Security Target specifies that the TOE does not support session resumption using session tickets; therefore, this conditional test does not apply.
Test Results	Pass
Execution Method	Manual

Test Case Number	055
SFR	FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
Test Objective	Test 1a [conditional]: If the TOE requires or can be configured to require a client certificate, the evaluator shall configure the TOE to require a client certificate and send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify that the handshake is not finished successfully and no application data flows.

	Test 1b [conditional]: If the TOE supports fallback authentication functions and these functions cannot be disabled. The evaluator shall configure the fallback authentication functions on the TOE and configure the TOE to send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify the TOE authenticates the connection using the fallback authentication functions as described in the TSS. Note: Testing the validity of the client certificate is performed as part of X.509 testing.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Verify that the TOE is configured to require a client certificate for remote TLS/HTTPS web administration. 2. Initiate a connection to the TOE web UI without sending a client certificate. 3. Verify that the TLS handshake is not finished successfully and that no application data flows. **Part 1b of this test assurance activity is not applicable as the ST selects "Not implement any administrator override mechanism" for this SFR.**
Test Results	The TOE fails to complete the TLS handshake and does not send any application data due to the missing client certificate from the remote endpoint. Part 1b of this test assurance activity is not applicable as the ST selects "Not implement any administrator override mechanism" for this SFR. - Pass
Execution Method	Manual

Test Case Number	056
SFR	FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
Test Objective	Test 2 [conditional]: If TLS 1.2 is claimed for the TOE, 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 connection is denied.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Initiate a connection to the TOE web UI while sending a valid client certificate. 2. Ensure that the certificate request message sent by the TOE to the client contains a "supported_signature_algorithm" list that does not contain an algorithm compatible with the client certificate. 3. Verify that the TLS handshake is not finished successfully.
Test Results	The evaluator observed that when the server's certificate request did not contain the correct supported signature algorithm and the client certificate was presented using the missing signature algorithm being used, the TOE correctly terminates the connection. - Pass
Execution Method	Manual

Test Case Number	057
SFR	FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
Test Objective	For all tests in this chapter the TLS client used for testing of the TOE shall support mutual authentication.

	Test 3: 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 correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not terminate in the claimed CA certificate). The evaluator shall verify that the attempted connection is denied.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Initiate a connection to the TOE web UI while sending a valid client certificate. 2. Verify that the TLS connection is successful, and that application data is exchanged. 3. From the test machine, generate a client identity certificate with an issuer field that identifies a CA recognized by the TOE as a trusted CA, but is signed using a different CA issuer key than the actual CA that is trusted by the TOE. 4. Initiate a connection to the TOE web UI while sending the client certificate that was generated in Step 3. Verify that the TLS connection is unsuccessful.
Test Results	The evaluator observed that the initial connection attempt was successful when a proper certificate chain was sent by the client. The evaluator also observed that the second connection attempt correctly failed when using a certificate signed by an imposter CA. - Pass
Execution Method	Manual

Test Case Number	058
SFR	FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
Test Objective	For all tests in this chapter the TLS client used for testing of the TOE shall support mutual authentication. Test 4: 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.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Initiate a connection to the TOE web UI while sending a valid client certificate, containing the Client Authentication purpose in the extendedKeyUsage field. 2. Verify that the TLS connection is successful, and that application data is exchanged. 3. Initiate a connection to the TOE web UI while sending a valid client certificate, without containing the Client Authentication purpose in the extendedKeyUsage field. Verify that the TOE does the connection.
Test Results	The evaluator observed that when the correct Client authentication purpose is specified in the extendedKeyUsage field the connection is successful. The evaluator also observed that when an incorrect purpose is assigned in the extendedKeyUsage field the TOE correctly terminates the connection. - Pass

Execution Method	Manual

Test Case Number	059
SFR	FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
Test Objective	For all tests in this chapter the TLS client used for testing of the TOE shall support mutual authentication. Test 5: The evaluator shall perform the following modifications to the traffic: a) Configure the server to require mutual authentication and then connect to the server with a client configured to send a client certificate that is signed by a Certificate Authority trusted by the TOE. The evaluator shall verify that the server accepts 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 (see RFC5246 Sec 7.4.8). The evaluator shall verify that the server rejects the connection. Note: Testing the validity of the client certificate is performed as part of X.509 testing
Test Instructions	Execute this test per the test steps.
Test Steps	**a)** 1. Initiate a connection to the TOE web UI while sending a valid client certificate. 2. Verify that the TLS connection is successful, and that application data is exchanged. **b)** 1. Initiate a connection to the TOE web UI while sending a valid client certificate. 2. Intercept and modify the traffic while in transit and modify a single byte in the signature block of the client's Certificate Verify handshake message. 3. Verify that the TOE does not establish the connection.
Test Results	The evaluator observed when mutual authentication is configured, and the presented client certificate's CA certificate has been imported and marked as trusted by the TOE, the TOE will correctly establish a connection. The evaluator also observed that when a byte in the signature block of the client certificate has been modified, the TOE correctly terminates the connection and audits the event. - Pass
Execution Method	Manual

Test Case Number	060
SFR	FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
Test Objective	The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows: Test 6: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Begin capturing packets between the TLS client web browser and the TOE.

	2. Initiate a connection from the TLS client web browser to the TOE. 3. Stop capturing packets between the TLS client web browser and the TOE. Verify that a trusted TLS channel using mutual authentication was established with the TOE.
Test Results	The evaluator observed that the mutual TLS connection was successful. - Pass
Execution Method	Manual

Test Case Number	061
SFR	FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
Test Objective	The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows: Test 7: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.
Test Instructions	Execute this test per the test steps.
Test Steps	Certificate validation failure after changing the presented certificate is tested in FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2 – Test 4 (Test Case 058). Additionally, all other types of certificate validation failures are tested between FCS_TLSS_EXT.2 and FIA_X509_EXT.1/Rev testing. The ST selects “Not implement any administrator override mechanism” for FCS_TLSS_EXT.2.2. As such, there are no “selected types of failure defined in the SFR”.
Test Results	Pass
Execution Method	Manual

Test Case Number	062
SFR	FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2
Test Objective	The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows: Test 8 [conditional]: The purpose of this test is to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – This conditional test does not apply because no override mechanism is defined for this SFR in the ST.
Test Results	Pass
Execution Method	Manual

Test Case Number	063

SFR	FCS_TLSS_EXT.2.3
Test Objective	The evaluator shall send a client certificate with an identifier that does not match an expected identifier and verify that the server denies the connection.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Initiate a connection to the TOE web UI while sending a valid client certificate, except that the value in the subject alternative name contains an IP address that is not the actual IP address used by the remote test client. 2. Verify that the TOE denies the connection.
Test Results	The evaluator observed that the TOE properly terminated the connection when a client certificate used a certificate with an improper CN/SAN entry (non-IP address). – Pass
Execution Method	Manual

3.3.3 Identification and Authentication

Test Case Number	064
SFR	FIA_AFL.1
Test Objective	The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g. any passwords entered as part of establishing the connection protocol or the remote administrator application): a) Test 1: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE (and, if the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall also use the operational guidance to configure the time period after which access is re-enabled). The evaluator shall test that once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful.
Test Instructions	Execute this test per the test steps.
Test Steps	**Remote CLI (SSH) [manual unlock by Security Administrator]:** 1. Authenticate to the TOE via the CLI as the Security Administrator. 2. Configure the number of successive failed authentication attempts to a value of "4" for each user. a. Navigate to "System Security Management" → "User Management". b. For each user, select "Edit". c. Specify the "Login Fail Count" value to "4". d. Select "Apply". 3. Attempt to authenticate to the TOE via SSH using a valid username and invalid password. 4. Verify the authentication is denied. 5. Repeat Steps 3 through 4, three more times. 6. Attempt to authenticate to the TOE via SSH using a valid username and a valid password. 7. Verify the authentication is denied due to lockout. 8. Repeat Steps 6 through 7 via the web GUI interface. 9. Authenticate to the TOE via the local console as a Security Administrator. 10. Manually unlock the locked account:

	a. Navigate to "System Security Management" → "User Management" → "Edit". b. Select "Unlock User". c. Select "OK". 11. Attempt to authenticate to the TOE via SSH with the previously locked account. 12. Verify authentication is successful. **Remote CLI (SSH) [elapsed time auto-unlock]:** 1. Authenticate to the TOE via the CLI as the Security Administrator. 2. Navigate to "System Security Management" → "Security Settings" → "Login". 3. Specify "Account Lockout Period" to a value of "60". 4. Select "Apply". 5. Configure the number of successive failed authentication attempts to a value of "5" for each user. a. Navigate to "System Security Management" → "User Management". b. For each user, select "Edit". c. Specify the "Login Fail Count" value to "5". d. Select "Apply". 6. Attempt to authenticate to the TOE via SSH using a valid username and invalid password. 7. Verify the authentication is denied. 8. Repeat Steps 3 through 4, four more times. 9. Attempt to authenticate to the TOE via SSH using a valid username and a valid password. 10. Verify the authentication is denied due to lockout. 11. Repeat Steps 6 through 7 via the web GUI interface. 12. Wait at least 60 seconds (until the account is auto unlocked). 13. Attempt to authenticate to the TOE via SSH with the previously locked account. 14. Verify authentication is successful. **Remote web GUI (TLS/HTTPS) [manual unlock by Security Administrator]:** 1. Authenticate to the TOE via the CLI as the Security Administrator. 2. Configure the number of successive failed authentication attempts to a value of "3" for each user. a. Navigate to "System Security Management" → "User Management". b. For each user, select "Edit". c. Specify the "Login Fail Count" value to "3". d. Select "Apply". 3. Attempt to authenticate to the TOE via the web GUI using a valid username and invalid password.

	4. Verify the authentication is denied. 5. Repeat Steps 3 through 4, two more times. 6. Attempt to authenticate to the TOE via the web GUI using a valid username and a valid password. 7. Verify the authentication is denied due to lockout. 8. Repeat Steps 6 through 7 via the SSH interface. 9. Authenticate to the TOE via the local console as a Security Administrator. 10. Manually unlock the locked account: a. Navigate to "System Security Management" → "User Management" → "Edit". b. Select "Unlock User". c. Select "OK". 11. Attempt to authenticate to the TOE via web GUI with the previously locked account. 12. Verify authentication is successful. **Remote web GUI (TLS/HTTPS) [elapsed time auto-unlock]:** 1. Authenticate to the TOE via the CLI as the Security Administrator. 2. Navigate to "System Security Management" → "Security Settings" → "Login". 3. Specify "Account Lockout Period" to a value of "120". 4. Select "Apply". 5. Configure the number of successive failed authentication attempts to a value of "7" for each user. a. Navigate to "System Security Management" → "User Management". b. For each user, select "Edit". c. Specify the "Login Fail Count" value to "7". d. Select "Apply". 6. Attempt to authenticate to the TOE via the web GUI using a valid username and invalid password. 7. Verify the authentication is denied. 8. Repeat Steps 3 through 4, six more times. 9. Attempt to authenticate to the TOE via the web GUI using a valid username and a valid password. 10. Verify the authentication is denied due to lockout. 11. Repeat Steps 6 through 7 via the SSH interface. 12. Wait at least 120 seconds (until the account is auto unlocked). 13. Attempt to authenticate to the TOE via the web GUI with the previously locked account. 14. Verify authentication is successful.
Test Results	The evaluator observed that the security administrator accounts that attempted to authenticate via the remote interfaces (remote CLI, remote Web GUI) are locked out for the configured time period or until a manual unlock action occurs when consecutive authentication failure attempts reach the configured limit for that interface. – Pass
Execution Method	Manual

Test Case Number	065
SFR	FIA_AFL.1
Test Objective	The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g. any passwords entered as part of establishing the connection protocol or the remote administrator application): b) Test 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows. If the administrator action selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall confirm by testing that following the operational guidance and performing each action specified in the ST to re-enable the remote administrator's access results in successful access (when using valid credentials for that administrator). If the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall wait for just less than the time period configured in Test 1 and show that an authorisation attempt using valid credentials does not result in successful access. The evaluator shall then wait until just after the time period configured in Test 1 and show that an authorisation attempt using valid credentials results in successful access.
Test Instructions	Execute this test per the test steps.
Test Steps	This assurance activity is performed in conjunction with FIA_AFL.1 - Test 1 (Test Case 064).
Test Results	Pass
Execution Method	Manual

Test Case Number	066
SFR	FIA_PMG_EXT.1
Test Objective	The evaluator shall perform the following tests. a) Test 1: The evaluator shall compose passwords that meet the requirements in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, and a minimum length listed in the requirement are supported and justify the subset of those characters chosen for testing.
Test Instructions	Execute this test per the test steps.

Test Steps	**NOTE: All characters claimed by the evaluation were tested by this test case.**
	
	a) CLI:
	
	1. Authenticate to the TOE via the CLI as the Security Administrator.
	2. Navigate to "System Security Management" → "User Management".
	3. Select a user and then select "Edit".
	4. Navigate to the "Password" tab.
	5. Specify the Current password.
	6. Specify a New Password with the following value:
	
	`abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTU`
	`VWXYZ1234567890!@#$%^().[]-+|~{}_`

	7. Confirm the new password. 8. Select "Apply". 9. Logout of the TOE. 10. Authenticate to the TOE via the CLI using the newly created password. 11. Verify the authentication is successful. 12. Repeat Steps 2 – 5. 13. Specify a New Password with the following value: `AB.cdef123456789!` 14. Repeat Steps 7 – 11.
Test Results	The evaluator observed that attempts to change the password to values compliant with the password length requirement of at least 15 characters and containing all of the claimed characters were successful. – Pass
Execution Method	Manual

Test Case Number	067
SFR	FIA_PMG_EXT.1
Test Objective	The evaluator shall perform the following tests. b) Test 2: The evaluator shall compose passwords that do not meet the requirements in some way. For each password, the evaluator shall verify that the TOE does not support the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that the TOE enforces the allowed characters and the minimum length listed in the requirement and justify the subset of those characters chosen for testing.
Test Instructions	Execute this test per the test steps.

Test Steps	**NOTE: All characters claimed by the evaluation were tested by this test case.**
	
	a) CLI:
	
	1. Authenticate to the TOE via the CLI as the Security Administrator.
	2. Navigate to "System Security Management" → "User Management".
	3. Select a user and then select "Edit".
	4. Navigate to the "Password" tab.
	5. Specify the Current password.
	6. Specify a New Password with the following value:
	
	`Abc123456789!@`
	
	7. Confirm the new password.
	8. Select "Apply".
	9. Verify the TOE does not accept the password change request.
	10. Repeat Steps 2 – 5.
	11. Specify a New Password with the following value:
	
	`!@#001Qa`
	
	12. Repeat Steps 7 – 9.

Test Results	The evaluator observed that attempts to change the password to values less than 15

	characters in length were unsuccessful. – Pass
Execution Method	Manual

Test Case Number	068
SFR	FIA_UAU.7
Test Objective	The evaluator shall perform the following test for each method of local login allowed: a) Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the local console as a Security Administrator. . 2. While entering password information, verify that the most obscured feedback is provided.
Test Results	The evaluator observed that the TOE does not echo the characters typed while a user is entering Password. – Pass
Execution Method	Manual

Test Case Number	069
SFR	FIA_UIA_EXT.1
Test Objective	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: a) Test 1: The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.
Test Instructions	Execute this test per the test steps.
Test Steps	**Local console (password based): As a Security Administrator…** 1. Authenticate to the TOE via the local console using a valid username and password. 2. Verify that the TOE successfully authenticated and that audit logs were generated reflecting the login. 3. Authenticate to the TOE via the local console using an invalid username and valid password. 4. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure. 5. Authenticate to the TOE via the local console using a valid username and an invalid password. 6. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure. 7. Authenticate to the TOE via the local console using an invalid username and an invalid password. 8. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure. **Web UI (password based): As a Security Administrator…**

	1. Authenticate to the TOE via the Web GUI using a valid username and password. 2. Verify that the TOE successfully authenticated and that audit logs were generated reflecting the login. 3. Authenticate to the TOE via the Web GUI using an invalid username and valid password. 4. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure. 5. Authenticate to the TOE via the Web GUI using a valid username and an invalid password. 6. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure. 7. Authenticate to the TOE via the Web GUI using an invalid username and an invalid password. 8. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure. **Remote SSH (password based): As a Security Administrator…** 9. Authenticate to the TOE via SSH using a valid username and password. 10. Verify that the TOE successfully authenticated and that audit logs were generated reflecting the login. 11. Authenticate to the TOE via SSH using an invalid username and valid password. 12. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure. 13. Authenticate to the TOE via SSH using a valid username and an invalid password. 14. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure. 15. Authenticate to the TOE via SSH using an invalid username and an invalid password. 16. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure. **Remote SSH (public/private key based):** 1. Authenticate to the TOE via SSH using a valid username and valid private key: `ssh ADMIN@<TOE-IP-Address> -i .\.ssh\id_ecdsa -o "PreferredAuthentications=publickey" -o "PasswordAuthentication=no" -o "PubkeyAuthentication=yes"` 2. Verify that the TOE successfully authenticated and that audit logs were generated reflecting the login.

	3. Authenticate to the TOE via SSH using an invalid username and a valid private key.
	
	`ssh <invaliduser>@<TOE-IP-Address> -i .\.ssh\id_ecdsa -o "PreferredAuthentications=publickey" -o "PasswordAuthentication=no" -o "PubkeyAuthentication=yes"`
	
	4. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure.
	5. Authenticate to the TOE via SSH using a valid username and an invalid private key (generate a new SSH keypair whose public key portion is not loaded into the TOE's authorized key file).
	
	`ssh ADMIN@<TOE-IP-Address> -i .\.ssh\id_ecdsa_invalid -o "PreferredAuthentications=publickey" -o "PasswordAuthentication=no" -o "PubkeyAuthentication=yes"`
	
	6. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure.
	7. Authenticate to the TOE via SSH using an invalid username and an invalid private key.
	
	`ssh <invaliduser>@<TOE-IP-Address> -i .\.ssh\id_ecdsa_invalid -o "PreferredAuthentications=publickey" -o "PasswordAuthentication=no" -o "PubkeyAuthentication=yes"`
	
	8. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure.

Test Results	The evaluator observed the TOE behaved correctly for all credential combinations for all interface/credential store combinations. The TOE also produced the correct audit records with the correct details. – Pass
Execution Method	Manual

Test Case Number	070
SFR	FIA_UIA_EXT.1
Test Objective	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: b) Test 2: The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.
Test Instructions	Execute this test per the test steps.
Test Steps	**The TOE does not allow for the configuration of any other functions other than the warning banner.**

	Remote CLI (password) 1. In a new SSH session, verify that the warning banner configured from the test Setup displayed prior to authentication to the TOE. 2. In a new SSH session, verify that no other services are available prior to authentication by entering a privileged command such as "ifconfig" as the username and password. 3. Verify that there is no other output apart from the expected output when invalid credential information is supplied at the authentication prompt and the configured FTA_TAB.1 warning banner. **Remote CLI (SSH key based authentication)** 1. In a new SSH session, verify that the warning banner configured from the test Setup displayed prior to authentication to the TOE. 2. In a new SSH session, verify that no other services are available prior to authentication by entering a privileged command such as "ifconfig" as the username and supplying a key file containing the string "ifconfig". 3. Verify that there is no other output apart from the expected output when invalid credential information is supplied at the authentication prompt and the configured FTA_TAB.1 warning banner. **Web UI (password based):** 1. In a new web UI session, verify that the warning banner configured in the Setup is displayed prior to authentication to the TOE. 2. In a new web UI session, verify that no other services are available prior to authentication by entering a privileged command such as "ifconfig" at the username and password prompts. 3. Verify that there is no other output apart from the expected output when invalid credential information is supplied at the authentication prompt and the configured FTA_TAB.1 warning banner.
Test Results	The evaluator observed that attempts to login with invalid credentials were unsuccessful for both SSH and GUI Console. The only item available prior to authentication on the system was the warning banner. – Pass
Execution Method	Manual

Test Case Number	071
SFR	FIA_UIA_EXT.1
Test Objective	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: c) Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.
Test Instructions	Execute this test per the test steps.
Test Steps	**The TOE does not allow for the configuration of any other functions other**

	than the warning banner. 1. In a new local console session, verify that the warning banner configured from the test Setup is displayed prior to authentication to the TOE. 2. In a new local console session, verify that no other services are available prior to authentication by entering a privileged command such as "ifconfig" as the username and password. 3. Verify that there is no other output apart from the expected output when invalid credential information is supplied at the authentication prompt and the configured FTA_TAB.1 warning banner.
Test Results	The evaluator observed that the warning banner was successfully configured and was displayed for the local CLI session. – Pass
Execution Method	Manual

Test Case Number	072
SFR	FIA_UIA_EXT.1
Test Objective	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: d) Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – The TOE is not a distributed TOE.
Test Results	Pass
Execution Method	Manual

Test Case Number	073
SFR	FIA_UAU_EXT.2
Test Objective	Evaluation Activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A – Per the assurance activity, evaluation activities for this requirement are covered under those for FIA_UIA_EXT.1
Test Results	Pass
Execution Method	Manual

Test Case Number	074
SFR	FIA_X509_EXT.1.1/Rev
Test Objective	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols:

	a) Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store). Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.
Test Instructions	Execute this test per the test steps.
Test Steps	**TOE acting as a TLS client connecting to a server:** 1. Create and install a server certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA certificates on the remote server. 2. Install the root CA, intermediate01 CA, and intermediate02 CA certificates into the TOE trust container. 3. Begin capturing packets between the server and the TOE. 4. Initiate a connection from the TOE to the server. 5. Stop capturing packets between the server and the TOE. 6. Validate connection is successful. 7. Remove the root CA certificate from the TOE CA trust container. 8. Repeat Steps 3-5. 9. Validate connection is unsuccessful. **TOE acting as a TLS server validating a TLS client certificate:** 1. Create and install a client certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA certificates on the remote client. 2. Install the root CA, intermediate01 CA, and intermediate02 CA certificates into the TOE trust container and designate them all as trust anchors. 3. Begin capturing packets between the client and the TOE. 4. Initiate a connection from the TOE to the client. 5. Stop capturing packets between the client and the TOE. 6. Validate connection is successful. 7. Remove the intermediate 01 CA certificate from the TOE CA trust container. 8. Repeat Steps 3-5. 9. Validate connection is unsuccessful. **TOE validation of signed software update:**

	1. Sign the software update using a code signing certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA certificates. 2. Install the root CA, intermediate01 CA, and intermediate02 CA certificates into the TOE trust container and designate them all as trust anchors. 3. Initialize the software update process. 4. Confirm that the TOE successfully validates the signed software update. 5. Remove the intermediate 01 CA from the trust container. 6. Initialize the software update process. 7. Confirm that the TOE fails to validate the signed software update.
Test Results	The evaluator observed that the TOE correctly accepts the presented certificate and successfully completes the connection when the CA certificates are present in the TOE's trust store. Additionally, the evaluator observed that the TOE correctly rejects the same presented certificate and terminates the connection when the intermediate 01 CA certificate was removed from the TOE's trust store. - Pass
Execution Method	Manual

Test Case Number	075
SFR	FIA_X509_EXT.1.1/Rev
Test Objective	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for selftesting is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols: Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.
Test Instructions	Execute this test per the test steps.
Test Steps	**TOE acting as a TLS client connecting to a server; TOE acting as a TLS server validating a TLS client certificate:** 1. Begin capturing packets between the server and the TOE. 2. Initiate a connection from the TOE to the server. 3. Stop capturing packets between the server and the TOE. **TOE validation of signed software update:** 1. Sign the software update using an expired code signing certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA certificates. 2. Install the root CA, intermediate01 CA, and intermediate02 CA certificates into the TOE trust container and designate them all as trust anchors. 3. Initialize the software update process. 4. Confirm that the TOE fails to validate the signed software update per as described in the TSS and the guidance documentation.

Test Results	The evaluator observed the TOE successful rejected an expired certificate and terminated the connection. – Pass
Execution Method	Manual

Test Case Number	076
SFR	FIA_X509_EXT.1.1/Rev
Test Objective	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for selftesting is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols: Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. 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. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.
Test Instructions	Execute this test per the test steps.
Test Steps	**CRL** **TOE acting as a TLS client connecting to a server; TOE acting as a TLS server validating a TLS client certificate:** 1. Load a valid server certificate onto the server. 2. Begin capturing packets between the server and the TOE as well as between the CRL distribution point and the TOE. 3. Initiate a connection from the TOE to the server. 4. Stop capturing packets between the server and the TOE as well as between the CRL distribution point and the TOE. 5. Load a revoked server certificate onto the server and re-prepare the CRL files as specified in the test Setup section. 6. Repeat Steps 2-4. 7. Load a valid server certificate onto the server. 8. Load a revoked intermediate01 CA certificate onto the server. 9. Repeat Steps 2-4. **TOE validation of signed software update:** 1. Sign the software update using a revoked code signing certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA

	certificates. 2. Install the root CA, intermediate01 CA, and intermediate02 CA certificates into the TOE trust container and designate them all as trust anchors. 3. Initialize the software update process. 4. Confirm that the TOE fails to validate the signed software update. 5. Sign the software update using a valid code signing certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA certificates. 6. Install the root CA, intermediate01 CA, and intermediate02 CA certificates into the TOE trust container and designate them all as trust anchors. 7. Generate a CRL on the CRL distribution point containing revocation information for the "intermediate01 CA" certificate. 8. Ensure that the TOE downloads the CRL generated in Step 7. 9. Initialize the software update process. 10. Confirm that the TOE fails to validate the signed software update.
Test Results	The evaluator confirmed that the TOE correctly accepts the unexpired presented certificate and successfully completes the connection when the CA certificates are present in the TOE's trust store. Additionally, the evaluator confirmed that the TOE correctly rejects the presented certificate and denies the connection when the presented certificate chain has a certificate that has been revoked. - Pass
Execution Method	Manual

Test Case Number	077
SFR	FIA_X509_EXT.1.1/Rev
Test Objective	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for selftesting is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols: Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.
Test Instructions	Execute this test per the test steps.
Test Steps	**CRL** **TOE acting as a TLS client connecting to a server; TOE acting as a TLS server validating a TLS client certificate:** 1. Place a CRL with no certificates revoked and signed by a CA that does not have the cRLsign key usage bit set at the CRL distribution point. 2. Initiate a connection from the TOE to the server (The connection will fail

	to succeed because of the invalid CRL). **TOE validation of signed software update:** 1. Sign the software update using a valid code signing certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA certificates, with the intermediate02 CA without the CRLsign key usage. 2. Install the root CA, intermediate01 CA, and intermediate02 CA certificates into the TOE trust container and designate them all as trust anchors. 3. Initialize the software update process. 4. Confirm that the TOE fails to validate the signed software update.
Test Results	The evaluator observed that when using a CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set the validation of the CRL correctly fails. – Pass
Execution Method	Manual

Test Case Number	078
SFR	FIA_X509_EXT.1.1/Rev
Test Objective	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for selftesting is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols: 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.)
Test Instructions	Execute this test per the test steps.
Test Steps	**TOE acting as a TLS client connecting to a server; TOE acting as a TLS server validating a TLS client certificate:** 1. Begin capturing packets between the TOE and the environmental entity. 2. Run the MITM tool to modify traffic. 3. Cause the TOE to initiate a connection to the environmental entity. 4. Stop capturing packets between the TOE and the environmental entity. 5. The connection will fail because the certificate will fail to parse correctly. **TOE validation of signed software update:** 6. Sign the software update using a valid code signing certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA certificates. 7. Modify the signature file certificate such that one of the bytes in the first eight bytes of the certificate is different than the original value. 8. Install the root CA, intermediate01 CA, and intermediate02 CA

	certificates into the TOE trust container and designate them all as trust anchors. 9. Initialize the software update process. 10. Confirm that the TOE fails to validate the signed software update.
Test Results	The evaluator observed that the TOE correctly fails to validate the certificate and denies the connection to the remote server when a single byte was modified in the first eight bytes of the presented certificate and terminates the connection. - Pass
Execution Method	Manual

Test Case Number	079
SFR	FIA_X509_EXT.1.1/Rev
Test Objective	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self testing is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols: Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC5280 Sec. 4.1.1.3), which is normally the last field in the certificate, and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
Test Instructions	Execute this test per the test steps.
Test Steps	**TOE acting as a TLS client connecting to a server; TOE acting as a TLS server validating a TLS client certificate:** 1. Begin capturing packets between the TOE and the environmental entity. 2. Run the MITM tool to modify traffic. 3. Cause the TOE to initiate a connection to the environmental entity. 4. Stop capturing packets between the TOE and the environmental entity. 5. The connection will fail because the certificate signature will fail to validate. **TOE validation of signed software update:** 6. Sign the software update using a valid code signing certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA certificates. 7. Modify the signature file certificate such that one of the bytes in the signatureValue field of the certificate is different than the original value. 8. Install the root CA, intermediate01 CA, and intermediate02 CA certificates into the TOE trust container and designate them all as trust anchors. 9. Initialize the software update process. 10. Confirm that the TOE fails to validate the signed software update.
Test Results	The evaluator observed that the TOE correctly fails to validate the certificate when a single byte in the presented certificate signatureValue field was modified and

	terminates the connection. – Pass
Execution Method	Manual

Test Case Number	080
SFR	FIA_X509_EXT.1.1/Rev
Test Objective	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self testing is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols: Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.)
Test Instructions	Execute this test per the test steps.
Test Steps	**TOE acting as a TLS client connecting to a server; TOE acting as a TLS server validating a TLS client certificate:** 1. Begin capturing packets between the TOE and the environmental entity. 2. Run the MITM tool to modify traffic. 3. Cause the TOE to initiate a connection to the environmental entity. 4. Stop capturing packets between the TOE and the environmental entity. 5. The connection will fail because the certificate hash will fail to validate. **TOE validation of signed software update:** 6. Sign the software update using a valid code signing certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA certificates. 7. Modify the signature file certificate such that one of the bytes in the public key of the certificate is different than the original value. 8. Install the root CA, intermediate01 CA, and intermediate02 CA certificates into the TOE trust container and designate them all as trust anchors. 9. Initialize the software update process. 10. Confirm that the TOE fails to validate the signed software update.
Test Results	The evaluator observed that the TOE correctly fails to validate the certificate when a single byte in the public key of the presented certificate is modified and terminates the connection. – Pass
Execution Method	Manual

Test Case Number	081
SFR	FIA_X509_EXT.1.1/Rev – TD0527
Test Objective	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary

	to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self testing is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols: The following tests are run when a minimum certificate path length of three certificates is implemented. Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall conduct the following tests: Test 8a: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain. Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid. Test 8c: The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.
Test Instructions	Execute this test per the test steps.
Test Steps	**TOE acting as a TLS client connecting to a server; TOE acting as a TLS server validating a TLS client certificate; TOE validation of signed software update:** **8a (Conditional) (applicable only to the FCS_TLSC_EXT.1 syslog interface):** 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). **8b (Conditional) (applicable only to the FCS_TLSC_EXT.1 syslog interface):** 6. 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 following command to generate the explicit parameter version of the key generated from using a named curve: `openssl ec -in <namedCurve.key> -param_enc explicit -out <explicit.key>` 7. 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. 8. Initiate a connection between the TOE and the remote endpoint. 9. Verify that the TOE treats the certificate chain as invalid (i.e. the connection is unsuccessful). **8c** 10. Load the EC "root CA" certificate onto the TOE's trust store. 11. Load the "int CA 01" certificate (that uses named curve EC parameters) that is signed by the EC "root CA" onto the TOE's trust store. 12. Verify that the TOE accepts the "int CA 01" certificate into the TOE's trust store. 13. Attempt to load the "int CA 01 explicit" certificate (that uses explicit format EC parameters) that is signed by the EC "root CA" onto the TOE's trust store. Verify that the TOE rejects the loading of the "int CA 01 explicit" certificate into the TOE's trust store.
Test Results	The evaluator observed that the TOE successfully validates a valid chain of EC certificates (terminating in a trusted CA certificate) is presented, where the elliptic

	curve parameters are specified as a named curve. The evaluator observed that the TOE correctly treats a certificate as invalid when a chain of EC certificates (terminating in a trusted CA certificate) is presented where the intermediate certificate uses an explicit format version of the Elliptic Curve parameters in the public key information field, is signed by the trusted EC root CA, and is valid in all other aspects. The evaluator observed that the TOE correctly treats a subordinate CA certificate as valid, where the elliptic curve parameters specifies a named curve, is signed by a trusted EC root CA, and is valid in all other aspects. The TOE successfully loaded the certificate into the trust store. Additionally, the evaluator confirmed that the TOE treats a subordinate CA certificate as invalid, where it specifies an explicit format version of the elliptic curve parameters, is signed by a trusted EC root CA, and is valid in all other aspects. The TOE correctly did not load the certificate into the trust store. - Pass
Execution Method	Manual

Test Case Number	082
SFR	FIA_X509_EXT.1/Rev
Test Objective	The evaluator shall perform the following tests for FIA_X509_EXT.1.2/Rev. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted. The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation). For each of the following tests the evaluator shall create a chain of at least three certificates: a self-signed root CA certificate, an intermediate CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain). a) Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains). The evaluator shall repeat these tests for each distinct use of certificates. Thus, for example, use of certificates for TLS connection is distinct from use of certificates for trusted updates so both of these uses would be tested. But there is no need to repeat the tests for each separate TLS channel in FTP_ITC.1 and FTP_TRP.1/Admin (unless the channels use separate implementations of TLS).
Test Instructions	Execute this test per the test steps.
Test Steps	**TOE acting as a TLS client connecting to a server; TOE acting as a TLS**

	server validating a TLS client certificate: 1. Present an otherwise valid intermediate02 CA certificate with one that does not contain the basicConstraints extension to the TOE. 2. Attempt to establish a connection to the remote server from the TOE. 3. Verify the connection attempt fails. **TOE validation of signed software update:** 1. Authenticate to the TOE web UI as the Security Administrator. 2. Navigate to "Node" → "Security" → "Certificate Authorities". 3. Define a new Certificate Authority. 4. Specify "Strict Base CRL (all)" for "CRL Method". 5. Specify "30 Minutes" for "CRL Update Interval". 6. Under "Certificate Validation Requirements" specify "ECDSA" for "Public Key Algorithm", specify "Required" for "Basic Constraints", "Extended Key Usage". 7. Click "Add". 8. Navigate to "Node" → "Security" → "Certificates & Keys". 9. Import the required CA certificates (including the intermediate02 CA certificate that does not contain the basicConstraints extension) and assign them to the Certificate Authority created in Steps 6 - 7. 10. Click "Add". 11. Verify that the TOE rejects the intermediate02 CA certificate.
Test Results	The evaluator observed that the TOE correctly rejects the certificate, as part of the validation of the leaf certificate belonging to the presented certificate chain, when the intermediate 02 CA in the presented chain does not contain the basicConstraints extension and terminates the connection. – Pass
Execution Method	Manual

Test Case Number	083
SFR	FIA_X509_EXT.1/Rev
Test Objective	The evaluator shall perform the following tests for FIA_X509_EXT.1.2/Rev. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted. The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation). For each of the following tests the evaluator shall create a chain of at least three certificates: a self-signed root CA certificate, an intermediate CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

	b) Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains). The evaluator shall repeat these tests for each distinct use of certificates. Thus, for example, use of certificates for TLS connection is distinct from use of certificates for trusted updates so both of these uses would be tested. But there is no need to repeat the tests for each separate TLS channel in FTP_ITC.1 and FTP_TRP.1/Admin (unless the channels use separate implementations of TLS).
Test Instructions	Execute this test per the test steps.
Test Steps	**TOE acting as a TLS client connecting to a server; TOE acting as a TLS server validating a TLS client certificate:** 1. Present an otherwise valid intermediate02 CA certificate with one that has the CA flag set to FALSE in the basicConstraints extension to the TOE. 2. Attempt to establish a connection to the remote server from the TOE. 3. Verify the connection attempt fails. **TOE validation of signed software update:** 1. Authenticate to the TOE web UI as the Security Administrator. 2. Navigate to "Node" → "Security" → "Certificate Authorities". 3. Define a new Certificate Authority. 4. Specify "Strict Base CRL (all)" for "CRL Method". 5. Specify "30 Minutes" for "CRL Update Interval". 6. Under "Certificate Validation Requirements" specify "ECDSA" for "Public Key Algorithm", specify "Required" for "Basic Constraints", "Extended Key Usage". 7. Click "Add". 8. Navigate to "Node" → "Security" → "Certificates & Keys". 9. Import the required CA certificates (including the intermediate02 CA certificate that does not contain the basicConstraints extension) and assign them to the Certificate Authority created in Steps 6 - 7. 10. Click "Add". 11. Verify that the TOE rejects the intermediate02 CA certificate.
Test Results	The evaluator observed that the TOE correctly rejects the certificate, as part of the validation of the leaf certificate belonging to the presented chain, when the intermediate 02 CA in the presented chain does not have the CA flag value set to TRUE and terminates the connection. – Pass
Execution Method	Manual

Test Case Number	084
SFR	FIA_X509_EXT.2
Test Objective	The evaluator shall perform the following test for each trusted channel: 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 guidance documentation to determine that all supported administrator-configurable options behave in their documented manner.
Test Instructions	Execute this test per the test steps.
Test Steps	**CRL** **TOE acting as a TLS client connecting to a server:** 1. Begin capturing packets between the TOE and the remote audit server. 2. Wait sufficient time for the TOE to attempt to retrieve CRLs from the CRL distribution point. 3. Initiate a connection from the TOE to the server. 4. Verify the connection succeeds. 5. Stop capturing packets. 6. Remove the "intermediate02.crl" from the CRL distribution point. 7. Begin capturing packets between the TOE and the remote audit server. 8. Wait sufficient time for the TOE to attempt to retrieve CRLs from the CRL distribution point. 9. Initiate a connection from the TOE to the server. 10. Verify the connection to the syslog server is denied due to the TOE being unable to verify the certificate (CRL unavailable). **TOE acting as a TLS server validating a TLS client certificate:** **Accept certificate:** 1. Create an "intermediate02.crl" that has a lifetime of 30 days. 2. Begin capturing packets between the TOE and the TLS client. 3. Wait sufficient time for the TOE to attempt to retrieve CRLs from the CRL distribution point. 4. Initiate a connection to the TOE from the TLS client. 5. Verify the connection succeeds. 6. Stop capturing packets. 7. Remove the "intermediate02.crl" from the CRL distribution point. 8. Begin capturing packets between the TOE and the TLS client. 9. Manually initiate a CRL update or wait sufficient time for the TOE to attempt to retrieve CRLs from the CRL distribution point. 10. Initiate a connection to the TOE from the TLS client. 11. Verify the connection to the TOE is accepted due to the TOE relying on the cached CRL obtained during Step 3. **Not accept certificate:** 1. Create an "intermediate02.crl" that expires within 15 minutes. 2. Begin capturing packets between the TOE and the TLS client. 3. Wait sufficient time for the TOE to attempt to retrieve CRLs from the CRL distribution point.

	4. Initiate a connection to the TOE from the TLS client. 5. Verify the connection succeeds. 6. Stop capturing packets. 7. Remove the "intermediate02.crl" from the CRL distribution point. 8. Begin capturing packets between the TOE and the TLS client. 9. Manually initiate a CRL update or wait sufficient time for the TOE to attempt to retrieve CRLs from the CRL distribution point. 10. Initiate a connection to the TOE from the TLS client. 11. Verify the connection to the TOE is denied due to the TOE being unable to verify the TLS client certificate. **TOE validation of signed software update:** 1. Sign the software update using a valid code signing certificate which chains to the root CA, intermediate01 CA, and intermediate02 CA certificates. 2. Install the root CA, intermediate01 CA, and intermediate02 CA certificates into the TOE trust container and designate them all as trust anchors. 3. Initialize the software update process. 4. Confirm that the TOE successfully validates the signed software update. 5. Remove the transferred update files from the TOE. 6. Remove the "intermediate02.crl" file from the TOE CRL cache. 7. Remove the "intermediate02.crl" file from the CRL distribution point. 8. Initialize the software update process. 9. Confirm that the TOE fails to validate the signed software update (CRL unavailable).
Test Results	The evaluator observed that the TOE accepted the certificate and successfully established the TLSC and TLS mutual authentication connections when the CRL is successfully downloaded and the certificates were not revoked. The evaluator observed that the TOE accepted the certificate and successfully established the TLSC and TLS mutual authentication connections when the CRL was not successfully downloaded when there was a previous non-expired cache of the CRL where the presented certificates were not revoked. The evaluator observed that the TOE rejected the certificate and successfully terminated the TLSC and TLS mutual authentication connections when the CRL was not successfully downloaded and there was no cache available to make a revocaion decision. The evaluator observed the TSF accepted the code signing certificate when the CRL was successfully downloaded and proceeded with the trusted update. The evaluator observed the TSF rejected the code signing certificate when the CRL was not successfully downloaded and halted the trusted update. - Pass
Execution Method	Manual

Test Case Number	108
SFR	FIA_X509_EXT.3
Test Objective	The evaluator shall perform the following tests: a) Test 1: The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated message

	and ensure that it conforms to the format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.
Test Instructions	Execute this test per the test steps.
Test Steps	**Public key, Common Name:** 1. Authenticate to the TOE via the web UI as the Security Administrator. 2. Navigate to "Node" → "Security" → "Certificates & Keys". 3. Under "Keys" choose "Add". 4. Select an available identifier. 5. Specify the Key Algorithm as "ECDSA". 6. Specify the Key Curve Name as "secp384r1". 7. Specify Common Name as "192.168.1.75". 8. Choose "Add". 9. Specify Renewal Mode as "Manual (CSR only)". 10. Select the newly created key corresponding to the identifier from Step 4 from the list of keys. 11. Under the "Key And Certificate Renewal" section, choose "Request". 12. Select and copy the CSR Data (PEM) from CSR Export. 13. On the Test Machine, verify the CSR data from Step 12 includes the ECDSA secp384r1 public key and "192.168.1.75" Common Name value: `openssl req -text -noout -verify -in <csr-from-Step12.pem>` 14. Verify that the CSR contains the expected public key type and size, and value for its Common Name.
Test Results	The evaluator observed that the certificate request was successfully generated and verified. - Pass
Execution Method	Manual

Test Case Number	109
SFR	FIA_X509_EXT.3
Test Objective	The evaluator shall perform the following tests: b) Test 2: The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message and demonstrate that the function succeeds.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Perform FIA_X509_EXT.3 – Test 1 (Test Case 108). 2. Using the CSR generated from Step 1, submit it to a CA for signing. 3. Upload the signed certificate issued by the CA into the TOE's certificate store, and do not associate it with any certification path. 4. Verify that the TOE fails to validate the signed certificate. 5. Associate the signed certificate that was uploaded in Step 3 with a valid certification path. 6. Verify that the TOE successfully validates the signed certificate.
Test Results	The evaluator observed that the TSF would not allow the CSR generated to be

	submitted for CA signing when there was not a valid certification path. When a valid certification path was provided, the TSF provided a Activate button which when selected validated the signed certificate. - Pass
Execution Method	Manual

3.3.4 Security Management

Test Case Number	085
SFR	FMT_MOF.1/ManualUpdate
Test Objective	The evaluator shall try to perform the update using a legitimate update image without prior authentication as Security Administrator (either by authentication as a user with no administrator privileges or without user authentication at all – depending on the configuration of the TOE). The attempt to update the TOE shall fail. The evaluator shall try to perform the update with prior authentication as Security Administrator using a legitimate update image. This attempt should be successful. This test case should be covered by the tests for FPT_TUD_EXT.1 already
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the web UI as a non-Security Administrator user. 2. Perform Steps 2 through 6 in FPT_TUD_EXT.1 – Test Case 093 to attempt to perform the update. 3. The second part of this test is already covered by testing performed in FPT_TUD_EXT.1 – Test Case 093.
Test Results	The evaluator was unsuccessful in initiating an update with using a user with limited privileges. The evaluator was successful in initiating an update with using a user with the security administrator role. - Pass
Execution Method	Manual

Test Case Number	See Test 88
SFR	FMT_MTD.1/CoreData
Test Objective	No separate testing for FMT_MTD.1/CoreData is required unless one of the management functions has not already been exercised under any other SFR.
Test Instructions	Execute this test per the test steps.
Test Steps	This test is satisfied by testing performing throughout the other test assurance activities.
Test Results	All functions were tested throughout the course of ATE testing. See Test 88 - Pass
Execution Method	Manual

Test Case Number	086
SFR	FMT_MTD.1/CryptoKeys
Test Objective	The evaluator shall try to perform at least one of the related actions (modify, delete, generate/import) without prior authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all). Attempts to perform related actions without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to manage cryptographic keys can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

	The evaluator shall try to perform at least one of the related actions with prior authentication as Security Administrator. This attempt should be successful.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the CLI as a non-Security Administrator user. 2. Navigate to "System Security Management" → "Public Key infrastructure (PKI)" → "Keys" → "Create private key". 3. Select one of the available "PKI_KEY-#" slots. 4. Verify that the selection does not advance the menu and that any further actions cannot be performed with respect to the keys. 5. Log out of the TOE. 6. Authenticate to the TOE via the CLI as the Security Administrator. 7. Repeat Steps 2 – 3. 8. Specify the Key Algorithm as "ECDSA". 9. Select "Next". 10. Specify the Key Curve Name as "secp384r1". 11. Select "Next", then "Next", then "Next", then "Next", and then "Apply".
Test Results	The evaluator was unsuccessful in generating a new private key with using a user with limited privileges. The evaluator successfully generated a new private key using a user with the security administrator role. - Pass
Execution Method	Manual

Test Case Number	087
SFR	FMT_SMF.1
Test Objective	The evaluator tests management functions as part of testing the SFRs identified in section 2.4.4. No separate testing for FMT_SMF.1 is required unless one of the management functions in FMT_SMF.1.1 has not already been exercised under any other SFR.
Test Instructions	Execute this test per the test steps.
Test Steps	This test is satisfied by testing performing throughout the other test assurance activities.
Test Results	Pass
Execution Method	Manual

Test Case Number	088
SFR	FMT_SMR.2 (and FMT_MTD.1/CoreData)
Test Objective	In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this cPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.
Test Instructions	Execute this test per the test steps.
Test Steps	This test is satisfied by testing performing throughout the other test assurance activities. This SFR assurance activity is satisfied by the testing of other SFRs in this test plan: • Ability to administer the TOE locally and remotely: (local administration) FIA_AFL.1, FIA_UAU.7, FIA_UIA_EXT.1, FTA_SSL_EXT.1,

	FTA_SSL.4, FTA_TAB.1; (remote administration) FCS_SSHS_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, FIA_AFL.1, FIA_PMG_EXT.1, FIA_UIA_EXT.1, FMT_MTD.1/CryptoKeys, FPT_TUD_EXT.1, FTA_SSL.3, FTA_SSL.4, FTP_TRP.1/Admin • Ability to configure the access banner: FTA_TAB.1 • Ability to configure the session inactivity time before session termination or locking: FTA_SSL_EXT.1, FTA_SSL.3 • Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates: FPT_TUD_EXT.1, FPT_TUD_EXT.2, FIA_X509_EXT.1/Rev • Ability to configure the authentication failure parameters for FIA_AFL.1: FIA_AFL.1 • Ability to manage the cryptographic keys: FMT_MTD.1/CryptoKeys • Ability to configure the cryptographic functionality: FMT_MTD.1/CryptoKeys, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, FCS_SSHS_EXT.1 • Ability to re-enable an Administrator account: FIA_AFL.1 • Ability to configure NTP: FCS_NTP_EXT.1, FPT_STM_EXT.1 • Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors: FIA_X509_EXT.1/Rev • Ability to import X.509v3 certificates to the TOE's trust store: FIA_X509_EXT.1/Rev • Ability to manage the trusted public keys database: FCS_SSHS_EXT.1
Test Results	The evaluator performed all functions throughout the course of ATE testing using both CLI and Web GUI administrator interfaces. - Pass
Execution Method	Manual

3.3.5 Protection of the TSF

Test Case Number	089
SFR	FPT_STM_EXT.1
Test Objective	The evaluator shall perform the following tests: a) Test 1: If the TOE supports direct setting of the time by the Security Administrator then the evaluator uses the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly. If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the web UI as the Security Administrator. 2. Navigate to "Node" → "General" → "Date & Time". 3. Specify "Disable" for NTP Operation. 4. Specify the Date [yyyy-mm-dd] and Time [hh:mm:ss] (in 24 hour time format) value to a value different than the current value. 5. Select "Apply". 6. Verify that the current TOE date/time value reflects the expected value.
Test Results	The evaluator observed that the TOE's clock was successfully configured to the specified value. - Pass

Execution Method	Manual

Test Case Number	090
SFR	FPT_STM_EXT.1
Test Objective	The evaluator shall perform the following tests: b) Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE and set up a communication path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the guidance documentation. If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.
Test Instructions	Execute this test per the test steps.
Test Steps	This test is performed as part of testing FCS_NTP_EXT.1.1 – Test Case 009
Test Results	Pass
Execution Method	Manual

Test Case Number	091 – TD0632
SFR	FPT_STM_EXT.1
Test Objective	c) Test 3: [conditional] If the TOE obtains time from the underlying VS, the evaluator shall record the time on the TOE, modify the time on the underlying VS, and verify the modified time is reflected by the TOE. If there is a delay between the setting the time on the VS and when the time is reflected on the TOE, the evaluator shall ensure this delay is consistent with the TSS and Guidance.
Test Instructions	Execute this test per the test steps.
Test Steps	N/A - Time is not obtained from VS.
Test Results	Pass
Execution Method	Manual

Test Case Number	092
SFR	FPT_TST_EXT.1
Test Objective	It is expected that at least the following tests are performed: a) Verification of the integrity of the firmware and executable software of the TOE b) Verification of the correct operation of the cryptographic functions necessary to fulfil any of the SFRs. Although formal compliance is not mandated, the self-tests performed should aim for a level of confidence comparable to: a) [FIPS 140-2], chap. 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. Note that the testing is not restricted to the cryptographic functions of the TOE. b) [FIPS 140-2], chap. 4.9.1, Cryptographic algorithm test for the verification of the correct operation of cryptographic functions. Alternatively, national requirements of any CCRA member state for the security evaluation of cryptographic functions should be considered as

	appropriate. The evaluator shall either verify that the self-tests described above are carried out during initial start-up or that the developer has justified any deviation from this. For distributed TOEs the evaluator shall perform testing of self-tests on all TOE components according to the description in the TSS about which self-test are performed by which component.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the CLI as the Security Administrator. 2. Select "Reboot NCU". 3. Confirm the reboot by selecting "OK". 4. Verify that the TOE performs an integrity check of the firmware and executable software of the TOE during its boot process. 5. Verify that the TOE verifies the correct operation of its cryptographic functionality during its boot process.
Test Results	The evaluator observed the TOE successfully performed the defined power-on self-tests (POST) for software integrity and cryptographic functionality. - Pass
Execution Method	Manual

Test Case Number	93
SFR	FPT_TUD_EXT.1
Test Objective	The evaluator shall perform the following tests: Test 1: The evaluator performs the version verification activity to determine the current version of the product. If a trusted update can be installed on the TOE with a delayed activation, the evaluator shall also query the most recently installed version (for this test the TOE shall be in a state where these two versions match). The evaluator obtains a legitimate update using procedures described in the guidance documentation and verifies that it is successfully installed on the TOE. For some TOEs loading the update onto the TOE and activation of the update are separate steps ('activation' could be performed e.g. by a distinct activation step or by rebooting the device). In that case the evaluator verifies after loading the update onto the TOE but before activation of the update that the current version of the product did not change but the most recently installed version has changed to the new product version. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again. If the verification of the hash value over the update file(s) against the published hash is not performed by the TOE, Test 3 shall be skipped. The evaluator shall perform Test 1, Test 2 and Test 3 (if applicable) for all methods supported (manual updates, automatic checking for updates, automatic updates). For distributed TOEs the evaluator shall perform Test 1, Test 2 and Test 3 (if applicable) for all TOE components.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the Web UI as the Security Administrator. 2. Record the Active Software TOE version by obtaining the version information by navigating to "Node" → "Software" → "NCU".

	3. Ensure the capability to transfer the update to the TOE via the Web UI upload form is enabled: a. Navigate to "Node" → "General" → "Controls" → "Functionality". b. Ensure "Upload & Download" is selected for "Local Computer Transfer". c. Click "Apply". 4. Perform the following steps to fetch and initiate the TOE software update: a. Navigate to "Node" → "Software" → "NCU" → "Transfer Software to Standby Area". b. Specify "Local Computer" for "Source Location". c. Click the "Import" button and select the following four files from the update package: E#######RC##.PGM F#######RC##.CON S#######RC##.PGM F#######RC##.SIG (For example, E7022022RC02.PGM, F7022022RC02.CON, S7022022RC02.PGM, F7022022RC02.SIG) d. After these files are imported, click "Transfer to Standby". 5. Prior to activation of update, confirm the TOE version corresponds to the current version by observing and notating the version information under the "Active Software Release" section of the "NCU" page. 6. Activate the most recently installed update by executing the following commands: a. On the "NCU" page, in the "Activate Software in Standby Area" section, choose "Activate". 7. After the TOE fully reboots, verify that the version number increased by repeating Steps 1-2 and comparing it to the version that was notated prior to the update.
Test Results	The evaluator observed that the TOE successfully shows the Active and Standby software releases. The TSF updated to the newer software version after the update was applied. The TOE's version verification activity confirmed the version increased as compared to the version reported prior to the update. - Pass
Execution Method	Manual

Test Case Number	94
SFR	FPT_TUD_EXT.1
Test Objective	The evaluator shall perform the following tests:

	Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates: 1) A modified version (e.g. using a hex editor) of a legitimately signed update 2) An image that has not been signed 3) An image signed with an invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate signature) 4) If the TOE allows a delayed activation of updates the TOE must be able to display both the currently executing version and most recently installed version. The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify, that both, current version and most recently installed version, reflect the same version information as prior to the update attempt. If the verification of the hash value over the update file(s) against the published hash is not performed by the TOE, Test 3 shall be skipped. The evaluator shall perform Test 1, Test 2 and Test 3 (if applicable) for all methods supported (manual updates, automatic checking for updates, automatic updates).
Test Instructions	Execute this test per the test steps.
Test Steps	**Modified version of valid signed update:** 1. Authenticate to the TOE via the Web UI as the Security Administrator. 2. Record the Active Software TOE version by obtaining the version information by navigating to "Node" → "Software" → "NCU". 3. Ensure the capability to transfer the update to the TOE via the Web UI upload form is enabled: a. Navigate to "Node" → "General" → "Controls" → "Functionality". b. Ensure "Upload & Download" is selected for "Local Computer Transfer". c. Click "Apply". 4. Perform the following steps to fetch and initiate the TOE software

update:

a. Navigate to "Node" → "Software" → "NCU" → "Transfer Software to Standby Area".
b. Specify "Local Computer" for "Source Location".
c. Click the "Import" button and select the following four files from the update package:

E#######RC##.PGM
F#######RC##.CON
S#######RC##.PGM
F#######RC##.SIG

(For example, E7022022RC02.PGM, F7022022RC02.CON, S7022022RC02.PGM, F7022022RC02.SIG)

d. After these files are imported, click "Transfer to Standby".

5. Verify that the invalid update fails to validate and that the update is not installed by confirming that the current version of the TOE corresponds to the version collected during Step 2.

Unsigned update:

6. Authenticate to the TOE via the Web UI as the Security Administrator.
7. Record the Active Software TOE version by obtaining the version information by navigating to "Node" → "Software" → "NCU".
8. Ensure the capability to transfer the update to the TOE via the Web UI upload form is enabled:

a. Navigate to "Node" → "General" → "Controls" → "Functionality".
b. Ensure "Upload & Download" is selected for "Local Computer Transfer".
c. Click "Apply".

9. Perform the following steps to fetch and initiate the TOE software update:

a. Navigate to "Node" → "Software" → "NCU" → "Transfer Software to Standby Area".
b. Specify "Local Computer" for "Source Location".
c. Click the "Import" button and select the following three files from the update package:

E#######RC##.PGM
F#######RC##.CON

S#######RC##.PGM

(For example, E7022022RC02.PGM, F7022022RC02.CON, S7022022RC02.PGM)

d. After these files are imported, click “Transfer to Standby”.

10. Verify that the invalid update fails to validate and that the update is not installed by confirming that the current version of the TOE corresponds to the version collected during Step 2.

Invalid signature:

11. Authenticate to the TOE via the Web UI as the Security Administrator.
12. Record the Active Software TOE version by obtaining the version information by navigating to “Node” → “Software” → “NCU”.
13. Ensure the capability to transfer the update to the TOE via the Web UI upload form is enabled:

    a. Navigate to “Node” → “General” → “Controls” → “Functionality”.
    b. Ensure “Upload & Download” is selected for “Local Computer Transfer”.
    c. Click “Apply”.

14. Perform the following steps to fetch and initiate the TOE software update:

    a. Navigate to “Node” → “Software” → “NCU” → “Transfer Software to Standby Area”.
    b. Specify “Local Computer” for “Source Location”.
    c. Click the “Import” button and select the following four files from the update package:

E#######RC##.PGM
F#######RC##.CON
S#######RC##.PGM
F#######RC##.SIG

(For example, E7022022RC02.PGM, F7022022RC02.CON, S7022022RC02.PGM, F7022022RC02.SIG)

d. After these files are imported, click “Transfer to Standby”.

15. Verify that the invalid update fails to validate and that the update is not installed by confirming that the current version of the TOE

	corresponds to the version collected during Step 2.
Test Results	The evaluator observed that the TOE correctly failed to update when invalid updates (modified binary via hex edit, missing signature, modified signature) were presented to the TOE. The TOE's active software version prior to the update attempts remained the same after the failed update attempts. - Pass
Execution Method	Manual

Test Case Number	95
SFR	FPT_TUD_EXT.1
Test Objective	The evaluator shall perform the following tests: Test 3 [conditional]: If the TOE itself verifies a hash value over an image against a published hash value (i.e. reference value) that has been imported to the TOE from outside such that the TOE itself authorizes the installation of an image to update the TOE, the following test shall be performed (otherwise the test shall be omitted. If the published hash is provided to the TOE by the Security Administrator and the verification of the hash value over the update file(s) against the published hash is performed by the TOE, then the evaluator shall perform the following tests. The evaluator first confirms that no update is pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. 1) The evaluator obtains or produces an illegitimate update such that the hash of the update does not match the published hash. The evaluator provides the published hash value to the TOE and calculates the hash of the update either on the TOE itself (if that functionality is provided by the TOE), or else outside the TOE. The evaluator confirms that the hash values are different, and attempts to install the update on the TOE, verifying that this fails because of the difference in hash values (and that the failure is logged). Depending on the implementation of the TOE, the TOE might not allow the user to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE. 2) The evaluator uses a legitimate update and tries to perform verification of the hash value without providing the published hash value to the TOE. The evaluator confirms that this attempt fails. Depending on the implementation of the TOE it might not be possible to attempt the verification of the hash value without providing a hash value to the TOE, e.g. if the hash value needs to be handed over to the TOE as a parameter in a command line message and the syntax check of the command prevents the execution of the command without providing a hash value. In that case the mechanism that prevents the execution of this check shall be tested accordingly, e.g. that the syntax check rejects the command without providing a hash value, and the rejection of the attempt is regarded as sufficient verification of the correct behaviour of the TOE in failing to verify the hash. The evaluator then attempts to install the update on the TOE (in spite of the unsuccessful hash verification) and confirms that this fails. Depending on the implementation of the TOE, the TOE might not allow to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE 3) If the TOE allows delayed activation of updates, the TOE must be able to display both the currently executing version and most recently installed version. The handling of version information of the most recently installed version might differ

	between different TOEs. Depending on the point in time when the attempted update is rejected, the most recently installed version might or might not be updated. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify, that both, current version and most recently installed version, reflect the same version information as prior to the update attempt If the verification of the hash value over the update file(s) against the published hash is not performed by the TOE, Test 3 shall be skipped. The evaluator shall perform Test 1, Test 2 and Test 3 (if applicable) for all methods supported (manual updates, automatic checking for updates, automatic updates).
Test Instructions	Execute this test per the test steps.
Test Steps	N/A - Per the test assurance activity, Test 3 is omitted because the verification of the update is not performed using a published hash.
Test Results	Pass
Execution Method	Manual

Test Case Number	107
SFR	FPT_TUD_EXT.2
Test Objective	The evaluator shall verify that the update mechanism includes a certificate validation according to FIA_X509_EXT.1 and a check for the Code Signing purpose in the extendedKeyUsage. The evaluator shall digitally sign the update with an invalid certificate and verify that update installation fails. The evaluator shall digitally sign the application with a certificate that does not have the Code Signing purpose and verify that application installation fails. The evaluator shall repeat the test using a valid certificate and a certificate that contains the Code Signing purpose and verify that the application installation succeeds. The evaluator shall use a previously valid but expired certificate and verifies that the TOE reacts as described in the TSS and the guidance documentation. Testing for this element is performed in conjunction with the assurance activities for FPT_TUD_EXT.1. The evaluator shall demonstrate that checking the validity of a certificate is performed at the time a certificate is used when performing trusted updates. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device.
Test Instructions	Execute this test per the test steps.
Test Steps	**Invalid update testing, including an update signed using an expired certificate that was previously valid is performed in FPT_TUD_EXT.1 – Test 2 (Test Case 094) and FIA_X509_EXT.1/Rev testing for TUD.** **Digitally signed update without "Code Signing" purpose:** 1. Authenticate to the TOE via the Web UI as the Security Administrator. 2. Record the Active Software TOE version by obtaining the version information by navigating to "Node" → "Software" → "NCU". 3. Ensure the capability to transfer the update to the TOE via the Web UI upload form is enabled: a. Navigate to "Node" → "General" → "Controls" →

	"Functionality". b. Ensure "Upload & Download" is selected for "Local Computer Transfer". c. Click "Apply". 4. Perform the following steps to fetch and initiate the TOE software update: a. Navigate to "Node" → "Software" → "NCU" → "Transfer Software to Standby Area". b. Specify "Local Computer" for "Source Location". c. Click the "Import" button and select the following four files from the update package: E#######RC##.PGM F#######RC##.CON S#######RC##.PGM F#######RC##.SIG (For example, E7022022RC02.PGM, F7022022RC02.CON, S7022022RC02.PGM, F7022022RC02.SIG) d. After these files are imported, click "Transfer to Standby". 5. Verify that the invalid update fails to validate and that the update is not installed by confirming that the current version of the TOE corresponds to the version collected during Step 3.
Test Results	The evaluator observed, that for all cases where incorrect certificates were presented to the TOE, the TOE correctly failed to update when invalid updates. The TOE's active software version prior to the update attempts remained the same after the failed update attempts. – Pass
Execution Method	Manual

3.3.6 TOE Access

Test Case Number	96
SFR	FTA_SSL_EXT.1
Test Objective	The evaluator shall perform the following test: a) Test 1: The evaluator follows the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the local console as a Security Administrator.. 2. Navigate to "System Security Management" → "Security Settings" → "Timeouts". 3. Specify the Craft Session Timeout value to 60 seconds.

	4. Select “Apply”. 5. Select “Cancel”. 6. Select “Quit”. 7. Authenticate to the TOE via the local console as a Security Administrator.. 8. Issue a command that invokes an audit record. 9. Leave the session idle for 60 seconds. 10. Verify that the TOE automatically terminated the session due to inactivity after 60 seconds of idle time has elapsed. 11. Repeat Steps 1 – 10, except replace the value of 60 in each step to a value of 90. 12. Repeat Steps 1-10, except replace the value of 60 in each step to a value of 120.
Test Results	The evaluator observed that for each configured inactivity timeout value, the TOE successfully terminated the local CLI session. - Pass
Execution Method	Manual

Test Case Number	97
SFR	FTA_SSL.3
Test Objective	For each method of remote administration, the evaluator shall perform the following test: a) Test 1: The evaluator follows the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a remote interactive session with the TOE. The evaluator then observes that the session is terminated after the configured time period.
Test Instructions	Execute this test per the test steps.
Test Steps	**Remote CLI (SSH):** 1. Authenticate to the TOE via the CLI as the Security Administrator. 2. Navigate to “System Security Management” → “Security Settings” → “Timeouts”. 3. Specify the Craft Session Timeout value to 180 seconds. 4. Select “Apply”. 5. Select “Cancel”. 6. Select “Quit”. 7. Authenticate to the TOE via SSH. 8. Perform activity that creates an audit record. 9. Leave the session idle for 180 seconds. 10. Verify that the TOE automatically terminated the session due to inactivity after 180 seconds of idle time has elapsed. 11. Repeat Steps 1 – 10, except replace the value of 180 in each step to a value of 240. 12. Repeat Steps 1-10, except replace the value of 180 in each step to a value of 300. **Remote web UI:** 13. Authenticate to the TOE via the CLI as the Security Administrator. 14. Navigate to “System Security Management” → “Security Settings” →

	"Timeouts". 15. Specify the Web Session Timeout value to 60 seconds. 16. Select "Apply". 17. Select "Cancel". 18. Select "Quit". 19. Authenticate to the TOE via the web UI as the Security Administrator. 20. Perform activity that creates an audit record. 21. Leave the session idle for 60 seconds. 22. Verify that the TOE automatically terminated the session due to inactivity after 60 seconds of idle time has elapsed. 23. Repeat Steps 13 – 22, except replace the value of 60 in each step to a value of 90. 24. Repeat Steps 13 – 22, except replace the value of 180 in each step to a value of 120.
Test Results	The evaluator observed that for each configured inactivity timeout value, the TOE successfully terminated the remote sessions for both the SSH CLI and Web GUI. – Pass
Execution Method	Manual

Test Case Number	98
SFR	FTA_SSL.4
Test Objective	For each method of remote administration, the evaluator shall perform the following tests: a) Test 1: The evaluator initiates an interactive local session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the local console as a Security Administrator.. 2. Select "Quit". Observe that the session has been terminated.
Test Results	The evaluator observed that the local administrator was successful in manually terminating the local CLI connection. – Pass
Execution Method	Manual

Test Case Number	99
SFR	FTA_SSL.4
Test Objective	For each method of remote administration, the evaluator shall perform the following tests: b) Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated.
Test Instructions	Execute this test per the test steps.
Test Steps	**Remote CLI (SSH):** 1. Authenticate to the TOE via SSH. 2. Select "Quit". 3. Observe that the session has been terminated. **Remote web UI:**

	1. Authenticate to the TOE via the web UI as the Security Administrator. 2. Select the username in the top right corner and then choose "Logout".
Test Results	The evaluator observed that the remote administrator was successful in manually terminating the remote SSH CLI and Web GUI connection. - Pass
Execution Method	Manual

Test Case Number	100
SFR	FTA_TAB.1
Test Objective	The evaluator shall also perform the following test: a) Test 1: The evaluator follows the guidance documentation to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.
Test Instructions	Execute this test per the test steps.
Test Steps	**Configure Banner:** 1. Authenticate to the TOE via the CLI as the Security Administrator. 2. Navigate to "System Security Management" → "Security Settings" → "Login" → "Access Warning". 3. In the Access Warning field, select "Enable". 4. In the Access Warning Message field, specify a message: "FTA_TAB.1 - WARNING" 5. Select "Apply". 6. Select "Cancel". 7. Select "Apply". **Local CLI** 1. Authenticate to the TOE via the local console as a Security Administrator.. 2. Confirm the specified text defined in the Setup is presented prior to authentication. **Remote CLI** 1. Authenticate to the TOE via SSH. 2. Confirm the specified text defined in the Setup is presented prior to authentication. **Web UI:** 1. Authenticate to the TOE via the web UI as the Security Administrator. 2. Confirm the specified text defined in the Setup is presented prior to authentication.
Test Results	The evaluator observed that the configured warning banner was displayed on all of the claimed interfaces used for authentication to the TOE (local console, SSH CLI, Web GUI). – Pass
Execution Method	Manual

3.3.7 Trusted Path/Channels

Test Case Number	101
SFR	FTP_ITC.1
Test Objective	The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report. The evaluator shall perform the following tests: Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful. Further assurance activities are associated with the specific protocols.
Test Instructions	Execute this test per the test steps.
Test Steps	**a) TOE and remote audit server** 1. Begin capturing packets between the TOE and the audit server. 2. On the TOE, perform an action that causes the TOE to initiate a connection to the audit server by performing an action that causes an audit record to be transmitted to the audit server. 3. Stop capturing packets between the TOE and the audit server. 4. Examine the packet capture and verify the data transmitted between the TOE and audit server are protected using TLS.
Test Results	The evaluator observed that the TOE successfully negotiated a secure channel to the audit server using TLS. Communications were not sent in plaintext to either server. - Pass
Execution Method	Manual

Test Case Number	102
SFR	FTP_ITC.1
Test Objective	The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report. The evaluator shall perform the following tests: Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE. Further assurance activities are associated with the specific protocols.
Test Instructions	Execute this test per the test steps.
Test Steps	Testing for this SFR is met by the testing performed in FTP_ITC.1 – Test 1 (Test Case 101).

Test Results	Pass
Execution Method	Manual

Test Case Number	103
SFR	FTP_ITC.1
Test Objective	The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report. The evaluator shall perform the following tests: Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext. Further assurance activities are associated with the specific protocols.
Test Instructions	Execute this test per the test steps.
Test Steps	Testing of this assurance activity is performed using FTP_ITC.1 – Test 1 (Test Case 101).
Test Results	Pass
Execution Method	Manual

Test Case Number	104
SFR	FTP_ITC.1
Test Objective	The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report. The evaluator shall perform the following tests: Test 4: Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route to the external IT entities. The evaluator shall, for each instance where the TOE acts as a client utilizing a secure communication mechanism with a distinct IT entity, physically interrupt the connection of that IT entity for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) a duration shorter than the application layer timeout but of sufficient length to interrupt the network link layer. The evaluator shall ensure that, when the physical connectivity is restored, communications are appropriately protected and no TSF data is sent in plaintext. In the case where the TOE is able to detect when the cable is removed from the device, another physical network device (e.g. a core switch) shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall not be performed at the virtual node (e.g. virtual switch) and must be physical in nature. Further assurance activities are associated with the specific protocols.

	For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of external secure channels to TOE components in the Security Target. The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public- facing document or report.
Test Instructions	Execute this test per the test steps.
Test Steps	**NOTE:** The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report. The developer response is as follows: TLS (syslog) – 16 minutes **TOE and remote audit server** 1. Establish a TLS network connection between the TOE and the remote audit server. 2. Begin capturing packets between the TOE and the audit server. 3. Physically disconnect the connection between the TOE and the audit server. 4. On the TOE, perform an action that causes the TOE to send audit records to the remote audit server. 5. After 16 minutes, restore physical connectivity between the TOE and the remote audit server. 6. Induce the transmission of audit data from the TOE to the remote audit server. 7. Stop capturing packets between the TOE and the audit server. 8. Examine the packet capture and verify the data transmitted between the TOE and audit server are protected using TLS. Repeat Steps 1-8, except in Step 5, replace “16 minutes” with “10 seconds”.
Test Results	The physical connection between the TOE and the remote entity was disconnected (at the network switch, such that network connectivity is physically connected between the TOE and the switch, but not between the switch and the remote entity). The evaluator observed that when physical connectivity is restored, communications are appropriately protected and no TSF data is sent in plaintext. - Pass
Execution Method	Manual

Test Case Number	105
SFR	FTP_TRP.1/Admin
Test Objective	The evaluator shall perform the following tests: Test 1: The evaluators shall ensure that communications using each specified (in

	the guidance documentation) remote administration method is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful. Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.
Test Instructions	Execute this test per the test steps.
Test Steps	Remote CLI (SSH): 1. Begin capturing packets between the TOE and the test machine. 2. Authenticate to the TOE via SSH. 3. Stop capturing packets between the TOE and the test machine. 4. Examine the packet capture and verify that the data transmitted between the test machine and the TOE is protected using SSH. 5. Refer to "FCS_SSHS_EXT.1.5 – Test Case 022" for the failure to establish a trusted path. Web UI (HTTPS/TLS): 1. Begin capturing packets between the TOE and the test machine. 2. Authenticate to the TOE via the web UI as the Security Administrator. 3. Stop capturing packets between the TOE and the test machine. 4. Examine the packet capture and verify that the data transmitted between the test machine and the TOE is protected using TLS. Refer to "FCS_TLSS_EXT.2 – Test Case 056" for the failure to establish a trusted path.
Test Results	The evaluator observed that the connection between the administrator workstation and the TOE successfully used SSH and TLS to access the CLI and Web GUI respectively. The evaluator confirmed that all channel data is not sent in plaintext. - Pass
Execution Method	Manual

Test Case Number	106
SFR	FTP_TRP.1/Admin
Test Objective	The evaluator shall perform the following tests: Test 2: The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext. Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.
Test Instructions	Execute this test per the test steps.
Test Steps	Testing for this SFR is met by the testing performed in FTP_TRP.1/Admin – Test 1 (Test Case 090).
Test Results	Pass
Execution Method	Manual

4 Evaluation Activities for SARs

This section addresses assurance activities that are defined in the *collaborative Protection Profile for Network Devices Version 2.2e* [NDcPP] that correspond with Security Assurance Requirements.

ADV_FSP.1-1 & **ADV_FSP.1-2** – *"The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant."*

This activity passes as Section 1.4 of the Security Target describes the purpose and method of use for each security relevant TSFI by enumerating all security relevant interfaces:

- E1: A direct local connection from the Terminal to the TOE via a serial or USB port. This connection is used for local administration of the TOE via a CLI.
- E2: A SSHv2 connection from the Remote Management Workstation to the TOE. This connection is used for remote administration of the TOE via a CLI.
- E3: A HTTPS connection from the Remote Management Workstation to the TOE. This connection is used for remote administration of the TOE via a Web GUI.
- E4: A TLS v1.2 trusted channel between the TOE and the external Audit Server used for external audit record storage.
- E5: A connection between the TOE and a Certificate Authority (CRL Distribution Point) used for X.509 certificate verification.
- E6: A connection between the TOE and an NTP server used as its time source. Note: the TOE can also be configured to use an internal clock as its time source.

The list also clearly identifies the interface that is out of scope for NDcPP testing:

- E7: FSP 3000R7's connection to the deployed network to provide its optical transport capabilities. While this connection is not part of the evaluated configuration, it is being included for completeness.

Each identified TSFI could be identified as to its functionality and the method of protection of the channels, when applicable.

ADV_FSP.1-3 – *"The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant."*

This activity passes as **t**he AGD was developed with the intent to provide the specific guidance for managing TOE functionality or a pointer to the necessary documentation as defined by the Intended Audience statement in Section 2. Thus, the evaluation team has determined that only the commands located within the AGD and the specific pointers to other documents are considered to be security relevant for this evaluation. Through the

completion of the independent functional testing, the evaluation team was able to test each SFR by executing the commands in each SFR's relevant test case(s). The evaluation team has determined that since the AGD document contains and/or provides the necessary pointer for all security relevant commands that were executed by the evaluation team in performing the independent testing, that the subset of the commands defined or referenced to in the AGD are all of the security relevant commands necessary to enforce the SFRs specified in the NDcPP.

ADV_FSP.1-5 – *"The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs."*

This activity passes as the TSFIs are labeled E1 through E6. The following list documents the SFR classes, how they are mapped to the TSFIs, and why the mapping is appropriate.

Security Audit (FAU_)
E1, E2, E3: These interfaces are used to perform management actions, such as configuring syslog connection, on the TOE. Each management action will generate an audit log with the identity of user. (FGEN.1, GEN.2, and STG_EXT.1)
E4: This interface is used for external audit storage via a Syslog server. (STG_EXT.1)
E6: This interface is used to keep time synchronized for accurate timestamps. (GEN.1, FAU_GEN.2, and STG_EXT.1)

Cryptographic Support (FCS_)
E1, E2, E3: Configuration of ciphers to support remote administration authentication and communications (password and public key) and TSF Data is sent over this interface and is protected with SSHv2. (SSHS_EXT.1, HTTPS_EXT.1, TLSS_EXT.1, TLSS_EXT.2, CKM.2, COP.1 as applicable to ciphers) CKM.2, FCS_COP.1 as applicable to ciphers)
E4: Audit data sent over this interface is protected by TLSv1.2 to the syslog server (TLSC_EXT.1, COP.1 as applicable to ciphers)
E5: Certificate revocation (CRL Distribution) checking is performed over this interface. (TLSC_EXT.1, TLSS_EXT.2)
E6: NTP server supporting NTPv4. (NTP_EXT.1 and COP.1 as applicable to ciphers)

Identification and Authentication (FIA_)
E1: Local interface does not echo password (UAU.7)
E1, E2, E3: Users of the TOE provide authentication credentials over these interfaces, subject to authentication failure handling, password policy, and password obfuscation. (UIA_EXT.1, UAU_EXT.2, AFL.1, PMG_EXT.1)
E5: Certificate revocation checking is performed over this interface. (X509_EXT.1 and X509_EXT.2)
E6: This interface is used to keep time synchronized for accurate inactivity timer and unlocking user accounts (AFL.1)

Security Management (FMT_)

E1, E2, E3: All management actions are performed over these interfaces. (SMF.1, SMR.1 MTD.1/CoreData, MTD.1/CryptoKeys, MOF.1/ManualUpdate)

Protection of the TSF (FPT_)
E1, E2, E3: All management actions are performed over these interfaces. (STM_EXT.1 APW_EXT.1, SKP_EXT.1, TST_EXT.1, TUD_EXT.1, TUD_EXT.2)

TOE Access (FTA_)
E1: All local user sessions are maintained over these interfaces and are subject to inactivity logouts, self-session termination, and display of audit banner. (SSL_EXT.1, SSL.4, TAB.1)
E1, E2, E3: All remote user sessions are maintained over these interfaces and are subject to inactivity logouts, self-session termination, and display of audit banner. (SSL.3, SSL.4, TAB.1)

Trusted Path/Channels (FTP_)
E2: Remote Administration data sent over this interface is protected with SSHv2 (TRP.1/Admin)
E3: Remote Administration data sent over this interface is protected with HTTPS/TLSv1.2 (TRP.1/Admin)
E4: Audit data sent over this interface is protected with SSHv2 (ITC.1)

AGD_OPE.1 – TD0536 *"The evaluator shall ensure that the Operational guidance is provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target."*

This activity passes as the TOE comes with its own set of administrative manuals that are clearly identified with the version of the TOE. When an end user purchases the TOE, they are given customer portal credentials for the pulling down of documentation and updates to ensure the user has access to the latest information. The *Adtran's FSP 3000R7 Network Element r22.2.2 Supplemental Administrative Guidance (AGD)* was developed with the intent to provide the specific guidance for installing, managing TOE functionality, and/or a pointer to the necessary documentation as defined by the Intended Audience. Tables 1 and 2 in the AGD and Tables 7 and 8 in the ST match and describe only the TOE models included in the evaluation and thus, the AGD addresses all platforms claimed by the evaluation. Thus, the evaluation team has determined that the AGD provides instructions for configuring and placing the TOE in its evaluated configuration in accordance with what is claimed in the Security Target.

"The evaluator shall ensure that the Operational guidance contains instructions for configuring any 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."

This activity passes as Section 6.1 of the AGD states the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality. The TOE is not subject to any situations that would prevent or delay key destruction and strictly conforms to the key destruction requirements.

"The evaluator shall ensure the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs."

This activity passes as Section 2 of the AGD states the FSP 3000R7 product, as a whole, provides a great deal of security functionality but only those functions that were in the scope of the claimed PP are discussed here. Any functionality that is not described in this supplemental document or in the FSP 3000R7 Network Element Security Target was not evaluated and should be exercised at the user's risk.

"In addition, the evaluator shall ensure that the following requirements are also met.

a) The guidance documentation shall contain instructions for configuring any 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.

This activity passes as Section 6.1 of the AGD states the administrator installing the TOE is expected to perform all of the operations in Sections 6.1 of this document. This will result in the TOE's cryptographic operations being limited to the claims made within the Common Criteria evaluation. There is no further configuration required on the TOE's cryptographic engine as these steps will limit the configuration (e.g., ciphersuites and algorithms) to those defined in the Security Target [1] as well as ensure automatic zeroization key destruction functionality. The TOE is not subject to any situations that would prevent or delay key destruction and strictly conforms to the key destruction requirements.
NOTE: The use of other cryptographic engines and cryptographic settings were not evaluated nor tested during the Common Criteria evaluation of the TOE

b) The documentation must describe the process for verifying updates to the TOE for each method selected for FPT_TUD_EXT.1.3 in the Security Target. The evaluator shall verify that this process includes the following steps:

5) 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).

This activity passes as Section 7.8.2 of the AGD states that the Security Administrator (Administrator or Provision) must download the TOE's update image from the Adtran Customer Portal page to the application server or local workstation. The administrator must use a computer separate from the TOE to recompute the hash of the downloaded image and verify it matches the published hash obtained from the Customer Portal page. Once this validation is complete, the administrator must sign the validated software, using the end user's approved code signing X.509v3 certificate. This creates the trusted update package. The trusted updated package is then placed on the customer's file server. The administrator must import the certificate authority (CA) certificates for the code signing certificate and mark the certificate as trusted.

6) Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes instructions that describe at least one method of validating the hash/digital signature.

This activity passes as Section 7.8 of the AGD covers the discussion of secure updates. This section provides an overview of how to obtain the updates and make them available to the TOE for installation and how the digital signature verification is done and what happens when the verification fails. Section 7.8 is then divided further subsections that provide clear instructions on how to display the current version, download the update, install the update using the CLI. The image will not be installed if the update fails to be verified and there is no administrative override.

c) The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the Evaluation Activities."

This activity passes as Section 2 of the AGD states this document references the Security Functional Requirements (SFRs) that are defined in the Security Target document and provides instructions on how to perform the security functions that are defined by these SFRs. The FSP 3000R7 product, as a whole, provides a great deal of security functionality but only those functions that were in the scope of the claimed PP are discussed here. Any functionality that is not described here or in the Adtran's FSP 3000R7 Network Element r22.2.2 Security Target was not evaluated and should be exercised at the user's risk." Section 7 reiterates this by stating, "The following sections provide information on managing TOE functionality that is relevant to the claimed Protection Profile.

AGD_PRE.1 – *"The evaluator shall examine the Preparative procedures to ensure they include a description of how the Security Administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target)."*

This activity passes as Section 5.3 of the AGD states defines a list of preparative procedures that provides the correct Operational Environment security objectives and administrative instructions for ensuring that they are satisfied.

"The evaluator shall examine the Preparative procedures to ensure they are provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target."

This activity passes as Section 5.3 of the AGD states defines a list of preparative procedures that map to the Operational Environment objectives defined in the ST.

"The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment."

This activity passes as Section 6.1 of the AGD provides step by step instructions to install and configure the TOE into the evaluated configuration. These steps have been verified during IND testing.

"The evaluator shall examine the preparative procedures to ensure they include instructions to manage the security of the TSF as a product and as a component of the larger operational environment."

This activity passes as Section 7 of the AGD is subdivided into specific sections that map to all of the security management functions defined in the ST.

"In addition, the evaluator shall ensure that the following requirements are also met.

The preparative procedures must

a) include instructions to provide a protected administrative capability; and
b) identify TOE passwords that have default values associated with them and instructions shall be provided for how these can be changed."

This activity passes as Section 7.3 of the AGD describes the RBAC enforcement mechanism and the specific roles that are considered to satisfy the Security Administrator role. Section 6.1 defines the only default password and that the TOE forces this default password to be changed upon first login.

ALC_CMC.1 – *"When evaluating that the TOE has been provided and is labelled with a unique reference, the evaluator performs the work units as presented in the CEM."*

The evaluation team verified that the Security Target (ST), TOE, and Supplemental Administrative Guidance (AGD) were labeled consistently to correctly identify the hardware and software versions in the CC evaluation. The ST clearly specifies the TOE Reference as being "Adtran's FSP 3000R7 Network Element operating with software release 22.2.2", which includes the following appliance models: SH1HU, SH7HU, and SH9HU. The TOE software version was shown to be 22.2.2 using the methods outlined in the AGD. The TOE hardware was identified by physical examination of the network appliance and the model number is on a sticker on the back.

ALC_CMS.1 – *"When evaluating the developer's coverage of the TOE in their CM system, the evaluator performs the work units as presented in the CEM."*

This activity passes as the evaluation team verified that the Security Target (ST), TOE, and Supplemental Administrative Guidance (AGD) were labeled consistently to correctly identify the hardware and software versions in the CC evaluation. The ST clearly specifies the TOE Reference as being "Adtran's FSP 3000R7 Network Element operating with software release 22.2.2", which includes the following appliance models: SH1HU, SH7HU, and SH9HU. The TOE software version was shown to be 22.2.2 using the methods outlined in the AGD. The TOE hardware was identified by physical examination of the network appliance and the model number is on a sticker on the back.

ATE_IND.1 – *"The evaluator performs the CEM work units associated with the ATE_IND.1 SAR. Specific testing requirements and EAs are captured for each SFR in Sections 2, 3 and 4.*

The evaluator should consult Appendix 709 when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation."

This activity passes as the evaluation team successfully performed the CEM work units associated with ATE_IND.1 SAR. There are not multiple variations of the TOE, but there are three models which are equivalent. Therefore, it is satisfactory to test one of the three models to obtain the assurance that the TOE is exactly conformant to the NDcPP.

AVA_VAN.1 – TD0547 – *"The evaluator shall examine the documentation outlined below provided by the developer to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously."*

"The evaluator formulates hypotheses in accordance with process defined in Appendix A. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3."

This activity passes as the evaluation team created a set of vulnerability tests to attempt to subvert the security of the TOE. These tests were created based upon the evaluation team's review of the vulnerability analysis evidence and independent research. The evaluation team conducted searches for public vulnerabilities related to the TOE. A few notable resources consulted include the cve.mitre.org and the nvd.nist.gov.

The following keywords were used individually and as part of various permutations and combinations to search for vulnerabilities identified in the public domain:

Keyword	Description
ADVA	This is a generic term for searching for known vulnerabilities produced by the acquired company as a whole.
Adtran	This is a generic term for searching for known vulnerabilities produced by the new acquiring company as a whole.
FSP3000/FSP 3000/FSP-3000	This is a generic term for searching for known vulnerabilities for the specific product.
SH1HU, SH7HU, SH9HU	These are the models for searching for known vulnerabilities for the specific product.
NCU-3/NCU3/NCU 3	This is a generic term searching for known vulnerabilities for the underlying operating system.
FSP Network Element	This is a generic term searching for known vulnerabilities for the underlying operating system.
Network Control Unit	This is a generic term searching for known vulnerabilities for the underlying operating system.
Libraries	
Numerous third party libraries were listed in a separately provided spreadsheet to the validators.*	These were specific third party libraries that are compiled into the TOE. Each library was research with the result catalogued in the separately provided. *The vendor has declared this list is not for public release.
Hardware	
T1042 (NXP QorIQ T-Series T1042E)	This is a generic term searching for known vulnerabilities for the TOE's underlying host processor.

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:

- Port Scanning

Remote access to the TOE should be limited to the standard TOE interfaces and procedures. This test attempted to find ways to bypass these standard interfaces of the TOE and open any other vectors of attack.

- Fuzzing – Mutated TYPE and CODE
  This attack attempts to determine if the TOE will properly handle malformed ICMP and IP packets that it receives with mutated TYPE and CODE values.

- Fuzzing – Mutated remaining field
  This attack attempts to determine if the TOE will properly handle malformed ICMP and IP packets that it receives with mutated remaining field values.

- Web Interface Vulnerability Identification (Nessus & Burp Suite)
  Burp Suite is a web application vulnerability assessment tool. It looks for major vulnerabilities including cross-site scripting, SQL injection, directory traversal, unchecked file uploads, etc. as well as less critical vulnerabilities such as unnecessary information disclosure. Nessus is a general-purpose network-based vulnerability scanner. It also looks for a suite of major vulnerabilities, including misconfigurations, default credentials, and web application related vulnerabilities.

The results of the tests were as follows:

The evaluation team conducted a public search on keywords and third party libraries pertaining to the TOE using the well-known vulnerability search sites such as National Vulnerabilities Database (NVD), Common Vulnerabilities and Exposures (CVE), U.S.-CERT, Tipping Point Zero Day Initiative, Offensive Security Exploit Database, Rapid7 Vulnerability Database, and Tenable. The public search was updated on March 16, 2024.

Additionally, the evaluation team performed penetration testing against the TOE at the Booz Allen CCTL facility in Laurel, MD in Sept through Oct 2023. All penetration testing attempts were properly repelled by the TOE and no vulnerabilities were found.

At the time of this report's submission, there were no known open vulnerabilities found pertaining to the TOE. There are currently no known discovered issues that could affect the security posture of a deployed system.

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

5 Conclusions

The evaluation team successfully applied all assurance activities defined in the NDcPP 2.2E and has concluded that the TOE and ST are in exact conformance to the NDcPP2.2E. The overall verdict for this evaluation is: Pass.

6 Glossary of Terms

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CA	Certificate Authority
CAVP	Cryptographic Algorithm Verification Program
CC	Common Criteria
CLI	Command-Line Interface
cPP	collaborative Protection Profile
CPU	Central Processing Unit
CRL	Certificate Revocation List
CSP	Content Security Policy
DRBG	Deterministic Random Bit Generator
HMAC	Hash-based Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
I&A	Identity and Access
IP	Internet Protocol
MAC	Message Authentication Code
NIAP	National Information Assurance Partnership
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OS	Operating System
OTH	Optical Transport Hierarchy
PP	Protection Profile
RAM	Random Access Memory
RBG	Random Bit Generator
RNG	Random Number Generator
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSH	Secure Shell
ST	Security Target
SVR	Server
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
UI	User Interface
WDM	Wavelength-Division Multiplexer

Table 6-1: Acronyms

Term	Definition
Administrator or 'Admin'	A user who is assigned the 'Admin' role on the TOE and has the ability to manage the TSF. Synonymous with Security Administrator.
Credential	Data that establishes the identity of a user (e.g., a cryptographic key or password).
Operating System (OS)	Software that manages hardware resources and provides services for applications.
Platform	A platform can be an operating system, hardware environment, a software-based execution environment, or some combination of these. These types platforms may also run atop other platforms.
Security Administrator	An authorized administrator role that is authorized to manage the TOE and its data. This TOE defines three separate user roles, but only the most privileged role (Admin) is authorized to manage the TOE's security functionality and is therefore considered to be the Security Administrator for the TOE.
Trusted Channel	An encrypted connection between the TOE and a system in the Operational Environment.
Trusted Path	An encrypted connection between the TOE and the application a Security Administrator uses to manage it (SSH client, terminal client, etc.).
User	In a CC context, any individual who has the ability to access the TOE functions or data.

Table 6-2: Terminology