Assurance Activities Report
for a Target of Evaluation

Gigamon GigaVUE Version 6.5

Assurance Activities Report (AAR)
Version 1.0

November 15, 2024

Evaluated by:
Booz | Allen | Hamilton

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

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

The Developer of the TOE:

Gigamon Inc.
3300 Olcott Street
Santa Clara, CA 95054

The Author of the Security Target:

Booz Allen Hamilton,
1100 West St.
Laurel, 20707 USA

The TOE Evaluation was sponsored by:

Gigamon Inc.
3300 Olcott Street
Santa Clara, CA 95054

Evaluation Personnel:

Herbert Markle
Rachel Kovach
Evan Seiz

Applicable Common Criteria Version

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

Common Evaluation Methodology Version

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

Table of Contents

1 Purpose

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

2 TOE Summary Specification Assurance Activities

The evaluation team completed the testing of the Security Target (ST) 'Gigamon GigaVUE Version 6.5 Security Target v1.0' 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.

The TOE is a standalone product and therefore the requirements for a distributed TOE are not applicable.

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 TSS states that the audit record contains the value that represents the key to identify the key for when generating/import of, changing, or deleting of cryptographic keys occurs. 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.

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.1 – *"The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally and how these records are protected against unauthorized modification or deletion. The evaluator shall ensure that the TSS describes the conditions that must be met for authorized deletion of audit records.*

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

The TSS states in section 8.1.3 that 8MB are allocated to local storage of audit logs. When a log file gets full it is rolled over to a backup file and compressed once it is full. A maximum of 8 log files exist and the oldest one is deleted by the rollover process whenever a new backup file is created. Audit logs cannot be modified by any role and only the Admin role can delete audit logs. 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.

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.

The TSS states in section 8.1.4 audit records are sent to a remote Syslog Server via an encrypted SSH channel over the Ethernet Management Port. It also states that the TOE is a standalone TOE that is responsible for storing its own audit records. When the Syslog Server is configured, the audit records are stored locally and immediately pushed to the Syslog Server. If Syslog Server connectivity is unavailable, audit records will only be stored locally. Upon re-establishment of communications with the Syslog Server, new audit records will resume being transmitted to it but the audit records that were generated during the time the Syslog Server connection was down remain stored locally and are not sent to the Syslog Server. New audit records are stored locally on the TOE under the /var/log directory in the file named “messages". The "message" file is archived when it reaches a specific size (8MB) by compressing it and saving the file as "messages.1.gz”. Meanwhile, a new "messages" file is created for new audit records and the other compressed messages files are rotated so that the 8 most recent compressed messages files are saved. The 8 compressed files are named "messages.1.gz”, “messages2.gz”, and so on. Therefore, as part of the file rotation “messages8.gz” will be deleted, "messages.7.gz" will be saved as "messages.8.gz", "messages.6.gz" will be saved as "messages.7.gz", and so on until the "messages" file is compressed into "messages.1.gz”. This mechanism guarantees a maximum limit of disk usage used by the log files. Only a user with the Admin role can delete the log files. 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.

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."*

The TSS states in section 8.2.1 that ECC keys using NIST curve P-256, P-384, P-521 are generated by the TOE in support of device authentication. This assurance activity is considered satisfied as the required information has been discovered.

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."

The TSS states in section 8.2.2 that the Elliptic Curve Diffie-Hellman (ECDH) key establishment scheme is used and the TOE complies with the NIST SP 800-56A Revision 3 key agreement scheme (KAS) primitives that are defined in section 5.6 of the SP. Additionally, the TSS states for TLS sessions the TOE can act as a TLS client and for SSH sessions the TOE can act as a SSH client and server as shown in the table below:

Scheme	**SFR**	**Service**
ECDH	FCS_TLSC_EXT.1.1	LDAP authentication
ECDH	FCS_TLSS_EXT.1.1	GigaVUE to Gigamon Fabric Manager connection
ECDH	FCS_SSHC_EXT.1.7	Audit server connection
ECDH	FCS_SSHS_EXT.1.7	CLI administration

This assurance activity is considered satisfied as the required information has been discovered.

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[1]). 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."

[1] Where keys are stored encrypted or wrapped under another key then this may need to be explained in order to allow the evaluator to confirm the consistency of the description of keys with the TOE functions.

The TSS in section 8.2.3 contains a table which specifies the key material, the origin, storage location, and how it is cleared. This covers SSH keys, authentication keys and TLS session keys. The TSS states that keys stored volatile memory are immediately zeroized using the function memset() upon deallocation. These keys are destroyed when sessions are closed. The TOE zeroizes all plaintext secret and private cryptographic keys in persistent storage by overwriting the file with zeroes and performing a read verify. Upon successful completion of the zeroization, the file is deallocated using the file system API unlink(). These keys are destroyed during import/re-installation or upgrade/regeneration. The TSS specifically states that there are no situations that would prevent or delay key destruction and strictly conforms to the key destruction requirements

The ST does not select “destruction by reference” or “invocation of an interface”. The TSS does not identify any keys stored in a non-plaintext form. The ST does not specify the use of “a value that does not contain any CSP”. This assurance activity is considered satisfied as the required information has been discovered.

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.”*

The TSS specifies in section 8.2.4 the encryption and decryption algorithms of AES-128 and AES-256 in both CBC and GCM modes. This assurance activity is considered satisfied as the required information has been discovered.

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.”*

The TSS specifies in section 8.2.5 the usage of ECDSA with a 256-bit key size and implements NIST P-256, P-384, and P-521 curves for signature generation and validation. This assurance activity is considered satisfied as the required information has been discovered.

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.”*

The TSS in section 8.2.6 lists which hash functions are used for data integrity, software integrity, TLS, digital signatures, and password hashing. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS specifies in section 8.2.7 for each hash function algorithm, the key length/size, digest size, block size, MAC output length, and the purpose/usage of the function (e.g. SSH, TLS). This assurance activity is considered satisfied as the required information has been discovered.

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

The TSS states in section 8.2.8 that the HTTPS implementation conforms to RFC 2818 and uses the TLS server implementations specified in FCS_TLSS_EXT.1. Since the HTTPS server does not enforce TLS mutual authentication, the only prerequisite to establishment of a TLS connection is that the peer initiates the communications. The TSS section provides a description list of how the TOE does or does not comply with each section of RFC 2818. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS states in section 8.2.9 that a CTR_DRBG is used. Two software-based entropy sources are used and the DRBG is seeded with a minimum of 256 bits of entropy. The TOE models uniformly provide two software-based entropy sources with estimated entropy output as described in the proprietary entropy specification. This assurance activity is considered satisfied as the required information has been discovered.

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

FCS_SSHC_EXT.1.2 – **TD0636** –*"The evaluator shall check to ensure that the TSS contains a list of the public key algorithms that are acceptable for use for user authentication and that this list is consistent with asymmetric key generation algorithms selected in FCS_CKM.1, hashing algorithms selected in FCS_COP.1/Hash, and signature generation algorithms selected in FCS_COP.1/SigGen. The evaluator shall confirm the TSS is unambiguous in declaring the TOE's ability to authenticate itself to a remote endpoint with a user-based public key.*

If password-based authentication method has been selected in the FCS_SSHC_EXT.1.2, then the evaluator shall confirm it is also described in the TSS."

The TSS specifies in section 8.2.10 the TOE's SSH client implementation only supports the use and generation of ecdsa-sha2-nistp384 algorithm for the public key user authentication. This list is consistent with the selections in FCS_CKM.1 which identifies the ECC scheme, FCS_COP.1/Hash which identifies SHA-256 and SHA-512, and FCS_COP.1.SigGen which identifies which has ECC with P-384 as one of its selections. When TOE acts as TLS client only public key-based user authentication is used when communicating with the remote audit server. This assurance activity is considered satisfied as the required information has been discovered.

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

The TSS specifies in section 8.2.10 that once a packet greater than 32,768 bytes is detected, the SSHv2 connection is dropped as described in RFC 4253. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_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."*

The TSS specifies in section 8.2.10 that AES-CBC-128, AES-CBC-256, aes128-gcm@openssh.com, and aes256-gcm@openssh.com are used for data encryption. This is consistent with the selections in the SFR. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_EXT.1.5 – **TD0636** – *"The evaluator shall confirm the TSS describes how a host-key public key (i.e., SSH server's public key) is associated with the server identity.*

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the host-key public key algorithms supported by the TOE are specified as well. The evaluator shall check the TSS to ensure that the host-key public key algorithms specified are identical to those listed for this component.

If x509v3-based public key authentication algorithms are claimed, the evaluator shall confirm that the TSS includes the description of how the TOE establishes the server's identity and how this identity is confirmed with the one that is presented in the provided certificate. For example, the TOE could verify that a server's configured IP address matches the one presented in the server's x.509v3 certificate."

The TSS specifies in section 8.2.10 the public key algorithms that are acceptable for host authentication are ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521 and will reject all others. This list is consistent with the selections in FCS_SSHC_EXT.1.5. The TOE does not support X509v3 certificates for the public key algorithm for authentication.

Additionally, the TSS specifies that the TOE's SSH client implementation will authenticate the identity of the audit server (i.e., SSH server) by using its local database (i.e., ~/.ssh/known_hosts) which associates each host name with its corresponding public key.

This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_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."*

The TSS specifies in section 8.2.10 that HMAC-SHA2-256 and HMAC-SHA2-512 are the supported data integrity algorithms. This is consistent with the selections in the SFR. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_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."*

The TSS specifies in section 8.2.10 that ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 are the key exchange methods and this is consistent with the SFR. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_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.

The TSS states in section 8.2.10 that the TOE has been hard coded to initiate a rekey when the session keys have been used for one hour (3600 seconds) or when 256 MB of data has been transmitted. Rekeying is performed upon reaching the threshold that is hit first. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_EXT.1.9 – This SFR does not contain any NDcPP TSS Assurance Activities.

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."

The TSS specifies in section 8.2.10 the SSH server implementation allows the use of ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 for public key user authentication. This list is consistent with the selections in FCS_COP.1/SigGen as ECC p-256, P-384, and P-521 are selected. This section also states that password-based authentication is also supported for the TOE acting as the SSH server for user authentication. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS states in section 8.2.10 that once a packet greater than 32,768 bytes is detected, the SSHv2 connection is dropped as described in RFC 4253. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS lists in section 8.2.10 AES-CBC-128, AES-CBC-256, aes128-gcm@openssh.com, and aes256-gcm@openssh.com for data encryption used. This is consistent with the selections in the SFR. This assurance activity is considered satisfied as the required information has been discovered.

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.*"*

The TSS specifies in section 8.2.10 that ecdsa-sha2-nistp384 as the only host public key algorithm and this is consistent with the selections in the SFR. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS specifies in section 8.2.10 that HMAC-SHA2-256, and HMAC-SHA2-512 as the supported data integrity algorithms. This is consistent with the selections in the SFR. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS specifies in section 8.2.10 that ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 are the key exchange methods and this is consistent with the SFR. This assurance activity is considered satisfied as the required information has been discovered.

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."

The TSS specifies in section 8.2.10 that the TOE has been hard coded to initiate a rekey when the session keys have been used for one hour or 1 GB when the TOE acts as a server. Rekeying is performed upon reaching the threshold that is hit first. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS specifies in section 8.2.11 that the cipher suites are:
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, and
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384.

This matches the selections in the SFR. This assurance activity is considered satisfied as the required information has been discovered.

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/application-configured 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."

The TSS specifies in section 8.2.11 that the presented identifier for the server certificate has to match the reference identifier in order to establish the connection. The TSS states the hostname reference identifier is the only supported value for X509 certificate validation. Wildcards cannot be defined as part of the reference identifier on the TOE, but the TOE will accept certificates with wildcards in the left-most label (e.g. *.example.com). The TOE supports the SAN extensions for certificate validation. The only Supported Elliptic Curves Extension included in the Client Hello are the NIST curves secp256r1, secp384r1, and secp521r1. Certificate pinning is not supported. When certificate validation fails, the connection is not established. Additionally, the TOE does not claim support for IP addresses for the reference identifier nor is the TOE distributed. Therefore, the TSS activities required for IP addresses and distributed TOE are not applicable. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS specifies in section 8.2.11 that the only supported elliptical curves included in the Client Hello are the NIST curves secp256r1, secp384r1, and secp521r1. This is not configurable and is therefore default behavior. This assurance activity is considered satisfied as the required information has been discovered.

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"*

The TSS specifies in section 8.2.11 that the TOE uses the TLS 1.2 protocol to secure the following connections and channels: LDAP server connection (TLS Client) used for authentication requests, and on HC and TA Series models only, a connection to a Gigamon Fabric Manager using HTTPS (TLS Server). When the TOE is operating in "Secure Cryptography Mode", TLS uses the following ciphersuites:

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

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. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS states in section 8.2.11 that the TOE will reject all connection attempts from TLS versions other than 1.2. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS specifies in section 8.2.11 that the TOE's ECDHE parameters are generated over NIST curves secp256r1, secp384r1, and secp521r. This assurance activity is considered satisfied as the required information has been discovered.

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.

The TSS states in section 8.2.11 that neither session resumption nor session tickets are supported by the TOE.

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)."

The TSS states in section 8.3.1 that the TSF uses a configurable counter for tracking consecutive failed authentication attempts and will lock an Admin or user account when the failure counter threshold is reached. A single counter is used for tracking a Remote CLI user's failed password-based and public-key based authentication attempts. A valid login that happens prior to the failure counter reaching its threshold will reset the counter to zero. When the failure counter threshold is reached, the offending account is locked and that user cannot login to the remote CLI until an administrative defined configurable time period is reached. Upon the configurable time period being reached, the TSF will reset the counter to zero and automatically unlock the account. The lockout duration is configurable (in seconds), with a default setting of 360 seconds. This assurance activity is considered satisfied as the required information has been discovered.

FIA_PMG_EXT.1- TD0792 – *“The evaluator shall check that the TSS lists the supported special character(s) and supported 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.”*

The TSS states in section 8.3.2 that passwords maintained by the TSF can be composed using any combination of upper case and lower case letters, numbers, and special characters including: “!”,”@”,”#”,”$”,”%”,”^”,”&”,”*”,”(“,”)”. The password policy is configurable by the Admin and supports the minimum password length of 8 characters to 30 characters. This assurance activity is considered satisfied as the required information has been discovered.

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

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."

The TSS states in section 8.3.3 users can authenticate to the TOE locally or remotely. Local users log in to the local console using a username and password via the Serial Port. Remote users can log in to the TOE via the remote CLI using username and password or SSH public key via the Ethernet Management Port. User authentication information that is sent remotely via the remote CLI is protected using SSHv2. When authenticating using username and password, these credentials are verified using either the TOE's local mechanism and credential repository or by an LDAP server that provides external authentication decisions. Valid credentials ensure a successful logon.

The TSS states in section 8.3.5 that the warning banner is displayed prior to the user authenticating. This is the only service prior to authentication. 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.

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."

The TSS states in section 8.3.6 certificate validity checking for outbound TLS connections to the LDAP Server. In addition to the validity checking that is performed by the TOE, the TSF will validate certificate revocation status using a certificate revocation list (CRL) that the TSF is configured to download automatically from a Certification Authority in the Operational Environment. In the event that the revocation status cannot be verified, the certificate will not be accepted.

The TSF validates certificates by ensuring that the certificate and the certificate path are valid in accordance with RFC 5280. In addition, the certificate path is terminated in a trusted CA certificate, the basicConstraints extension is present, and the CA flag is set to TRUE for all CA certificates. The TSF also ensures that the extendedKeyUsage field includes the correct purpose for its intended use. This includes Server Authentication for TLS server certificates. The TSF certificate validation does not support TLS client certificates, certificates associated with OCSP responses, or code signing certificates.

Revocation checking is handled the same whether a full certificate chain or a leaf certificate is presented.

This assurance activity is considered satisfied as the required information has been discovered.

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."

The TSS states in section 8.3.6 that the Security Administrator must import the root certificate of the remote server for certificate validation. Additionally, this section states that the Security Administrator has the ability to generate or import a certificate/certificate chain generated off-TOE for use as the TOE's server certificate for the TLS Server functionality.

The Administrative Guidance provides detailed steps to configure the TOE to use the correct root certificates for validating an LDAP server (section 7.1.2). The guidance also provides the steps necessary to configure the TOE with the correct certificates to send to an external TLS client device (Gigamon Fabric Manager). This functionality is covered under section 6.9.2 of the AGD.

The TSS description states that the TOE performs revocation checking using certificate revocation lists (CRL). In the event that the revocation status cannot be verified, the certificate will not be accepted.

The only distinctions between trusted channels is when the TOE acts as a TLS client or TLS Server. Both functionalities are covered in section 8.3.6 of the ST and in section 7.1.2 and 6.92 of the AGD. Therefore, this assurance activity is considered satisfied as the required information has been discovered.

This assurance activity is considered satisfied as the required information has been discovered.

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."*

The ST author did not select "device-specific information" and therefore these requirements are not applicable.

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

The TOE is a standalone product and therefore the requirements for a distributed TOE are not applicable.

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."

The TSS states in section 8.4.2 that the only available functionality prior to administrator authentication is the display of the warning banner. The TSF uses role-based access control to assign each user account to one or more roles. Only the Admin role is authorized to perform the management functions associated with the TSF. This is consistent with the administrative guidance which does not identify any configuration functions available to any user prior to authentication. This assurance activity is considered satisfied as the required information has been discovered.

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

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 8.4.3 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 commands. Within the TSF, this behavior is limited to the generation and import/removal of X.509 certificates, and the generation, import and deletion of SSH keys. 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."

The TSS specifies in section 8.4.4 the management functions and identifies which management functions are available through the Local console and Remote CLI interfaces. The ST defined management functions align with those discovered in the guidance document and testing. 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_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."*

The TSS states in section 8.4.5 that the security management function available to authorized users of the TOE are mediated by a role-based access control system. The role-based access control system consists of the Admin and Monitor role and is enforced via all the interfaces: local console and remote CLI. All SFR relevant management activity is performed by the Admin while the Monitor role only provides view-only access to ports and configurations. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS states in section 8.5.1 that all passwords are stored hashed by SHA-512. The password file cannot be viewed by any user on the TOE regardless of the user's role. This assurance activity is considered satisfied as the required information has been discovered.

FPT_SKP_EXT.1 – *"The evaluator shall examine the TSS to determine that it details how any pre-shared 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."*

The TSS states in section 8.5.2 that public keys are stored in the configuration database which is integrity checked at boot time. Secret and private keys are stored in plaintext on the hard drive but cannot be accessed by any user via any interface. This assurance activity is considered satisfied as the required information has been discovered.

FPT_STM_EXT.1 – **TD0632** – *"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.*

If "obtain time from the underlying virtualization system" is selected, the evaluator shall examine the TSS to ensure that it identifies the VS interface the TOE uses to obtain time. If there is a delay between updates to the time on the VS and updating the time on the TOE, the TSS shall identify the maximum possible delay."

The TSS states in section 8.5.3 that the TOE has an underlying hardware clock that is used for keeping time. A user with the Admin role can configure the time manually. The TOE uses time data for audit record timestamps, inactivity timeout for administrative sessions, expiration checking of certificates and timer for lockout duration as described in FIA_AFL.1. The TOE does not obtain time from an underlying virtualization system. This assurance activity is considered satisfied as the required information has been discovered.

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."

The TSS states in section 8.5.4 that all binaries are in a read-only partition which prevents modification. The TOE has a configuration database that is integrity checked at boot time. The udiag performs a suite of power-on self-tests on the major components of the TOE including memory, CPU, UART, Ethernet controllers, transceivers). The TSS states that pci_diag is run, which checks components connected to PCIe interfaces. If the TOE fails any of the integrity checks such as cryptography or software integrity, the platform will be placed into a safe mode. In safe mode, the device will operate in a limited manner which requires user intervention to bring the appliance back into a normal state after fixing the issues. The console display clearly indicates that the appliance is in

SAFE mode along with the diagnostic information. 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.

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."

The TSS states in section 8.5.5 that for the local console and CLI the user can run show version. The TSS states that an image is installed on the inactive partition and in order for it to be activated the TOE must boot off the inactive partition. The TSS details that in order to update the TOE, the Admin will access a Gigamon-hosted site and enter a username and password to download the image to their local machine. After downloading the image, the Admin will fetch the image through the remote CLI. A digital signature check is made prior to installation of the package. If the verification is successful, the image will be installed; otherwise, the Admin will receive an error message and the image will not be installed. Automatic update options have not been selected. The TOE does not support publish hash and 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.

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."*

The TSS states in section 8.6.1 that the TOE is designed to terminate a local session after a specific period of time. The default setting is 15 minutes and it is configurable by an Admin. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS states in section 8.6.2 that the TOE can be configured to terminate remote interactive sessions via Remote SSH and local console CLIs with a configurable value of 0 (no timeout), or between .25 and 35791 minutes. This assurance activity is considered satisfied as the required information has been discovered.

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

The TSS states in section 8.6.3 that the Admin is able to terminate their own session by entering the "exit" command when logged into the local console or remote CLI. This assurance activity is considered satisfied as the required information has been discovered.

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)."*

The TSS details in section 8.6.4 that the two possible ways to authenticate to the TOE are the local console and remote CLI. Each method of access has a configurable login banner that can be shown prior to authentication. This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS identifies in section 8.7.1 channels to following external entities: LDAP via TLS v1.2, Syslog via SSHv2, Gigamon Fabric Manager via HTTPS.

For TLS and SSH: In the evaluated configuration, the TOE connects with an audit server using SSHv2 (FCS_SSHC_EXT.1) to encrypt the audit data that traverses the channel. When remote authentication is configured, the TOE connects to an LDAP Server using TLS v1.2 (FCS_TLS_EXT.1) to send authentication requests for a user attempting to login to the local console or remote CLI. These remote endpoints are authenticated using TLS server certificates and SSH host keys. In each of these instances, the TOE initiates communication as the client using the cryptographic protocol in the manner described by their respective SFRs. These protocols are used to protect the data traversing the channel from disclosure and/or modification.

For HTTPS: As part of establishing the HTTPS (FCS_HTTPS_EXT.1) connection, the TOE confirms the identity of the Gigamon Fabric Manager by validating the credentials supplied during the connection establishment.

This assurance activity is considered satisfied as the required information has been discovered.

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."*

The TSS states in section 8.7.2 that remote administration is performed via a CLI that is protected by SSHv2 which is consistent with the protocol claims made by the ST. This assurance activity is considered satisfied as the required information has been discovered.

3 Operational Guidance Assurance Activities

The evaluation team completed the testing of the Operational Guidance, which includes the review of the *Gigamon GigaVUE Version 6.5 Supplemental Administrative Guidance* *v1.0* (AGD) document and confirmed that the Operational Guidance contains all Assurance Activities as specified by the *collaborative Protection Profile for Network Devices V2.2e* [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. The AGD includes references to other guidance documents that must be used to properly install, configure, and operate the TOE in its evaluated configuration. The AGD and its references to other Gigamon GigaVUE guidance documents were reviewed to assess the Operational Guidance Assurance Activities. The AGD contains references to these documents in Chapter 4 and these references can also be found below:

[1] Gigamon GigaVUE Version 6.5 Security Target, v1.0 [ST]
[2] GigaVUE-OS CLI Reference Guide, GigaVUE-OS, v1.0, Product Version 6.5, Document Version 1.0
[3] GigaVUE-HC1 Hardware Installation Guide, GigaVUE H Series, v1.0 Product Version 6.5, Document Version 1.0
[4] GigaVUE-HC1-Plus Hardware Installation Guide, GigaVUE H Series, v1.0 Product Version 6.5, Document Version 1.0
[5] GigaVUE-HC3 Hardware Installation Guide, GigaVUE H Series, v1.0 Product Version 6.5, Document Version 1.0
[6] GigaVUE-HCT Hardware Installation Guide, GigaVUE H Series, v1.0 Product Version 6.5, Document Version 1.0
[7] GigaVUE TA25 Hardware Installation Guide, GigaVUE TA Series, v1.0 Product Version 6.5, Document Version 1.0
[8] GigaVUE TA25E Hardware Installation Guide, GigaVUE TA Series, v1.0 Product Version 6.5, Document Version 1.0
[9] GigaVUE TA200 Hardware Installation Guide, GigaVUE TA Series, v1.0 Product Version 6.5, Document Version 1.0
[10] GigaVUE TA200E Hardware Installation Guide, GigaVUE TA Series, v1.0 Product Version 6.5, Document Version 1.0
[11] GigaVUE TA400 Hardware Installation Guide, GigaVUE TA Series, v1.0 Product Version 6.5, Document Version 1.0
[12] GigaVUE G-TAP A Series 2 Hardware Installation Guide, G-TAP A-TX21, G-TAP A-TX21-C, G-TAP A-SF21, v1.0 Product Version 6.5, Document Version 1.0

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 an example of an audit record 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 Gigamon GigaVUE-OS Version 6.5. 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 Gigamon GigaVUE-OS Version 6.5 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 Gigamon GigaVUE's standard documentation set for the product which contains functionality that is outside the scope of the evaluation. The GigaVUE 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 Gigamon GigaVUE Version 6.5 Security Target was not evaluated and should be exercised at the user's risk."
This assurance activity is considered satisfied as the required information has been discovered.

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.1 – *"The evaluator shall examine the guidance documentation to determine that it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion."*

Section 8.1 of the AGD states that users of any role can view audit log files, but only Admin users can delete audit log files. No modification of log files is permitted, regardless of role. Users with the Admin role are considered trusted users and are not expected to delete audit records. This assurance activity is considered satisfied as the required information has been discovered.

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 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 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."

Section 8.1 of the AGD, and its subsections, describes how to configure the TOE syslog client to securely transmit audit records the TOE generates to a remote syslog server via SSH. The AGD states that audit records are stored both locally and also sent immediately to the audit server over an SSH encrypted channel. Upon re-establishment of communications with the audit server, new audit records will resume being transmitted but the audit records that were generated during the time the audit server connection was down remain stored locally and are not sent to the audit server.

Section 8.1 of the AGD describes the behavior for the handling of "when the local storage space for audit data is full" configuration option chosen for FAU_STG_EXT.1.3. The description provided in the AGD states "New audit records are stored locally on the TOE under the /var/log directory in the file named "messages". The "message" file is archived when it reaches a specific size (8 MB) by compressing it and saving the file as "messages.1.gz". Meanwhile, a new "messages" file is created for new audit records and the other compressed messages files are rotated so that the 8 most recent compressed messages files are saved. The 8 compressed files are named "messages.1.gz", "messages2.gz", and so on. Therefore, as part of the file rotation "messages8.gz" will be deleted, "messages.7.gz" will be saved as "messages.8.gz", "messages.6.gz" will be saved as "messages.7.gz", and so on until the "messages" file is compressed into "messages.1.gz”. This mechanism guarantees a maximum limit of disk usage used by the log files.” This assurance activity is considered satisfied as the required information has been discovered.

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.”*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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).”*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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).”

Section 6.3 of the AGD specifically states that automatic zeroization key destruction functionality is default behavior for the TOE. “The TOE is not subject to any situations that would prevent or delay key destruction and strictly conforms to the key destruction requirements.” This is consistent with Section 8.2.3 of the ST which also specifically states: “The TOE is not subject to any situations that would prevent or delay key destruction and strictly conforms to the key destruction requirements.” This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

FCS_COP.1.1/Hash – *"The evaluator checks the AGD documents to determine that any configuration that is required to configure the required hash sizes is present."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 6.9.2 of the AGD provides the steps to configure the TOE to use encrypted communication including TLS/HTTPS for connections with the Gigamon Fabric Manager Server. These steps include the generation of a server certificate and steps to import a certificate chain that has been generated off TOE. This assurance activity is considered satisfied as the required information has been discovered.

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

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_EXT.1.1 – There are no NDcPP AGD assurance activities for this SFR.

FCS_SSHC_EXT.1.2 – **TD0636** – *"The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections initiated by the TOE."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_EXT.1.3 – There are no NDcPP AGD assurance activities for this SFR.

FCS_SSHC_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)."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_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)."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_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)."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. Section 6.5 of the AGD states that the MAC algorithms defined in the ST are the only ones included in the evaluated configuration and that the "none" MAC algorithm is never allowed for SSH. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_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."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_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."*

Section 6.5 of the AGD states that the SSH session key thresholds for time and amount of transmitted data are not configurable in the evaluated configuration. It also states that whichever threshold (traffic or time) occurs first is when the TOE will initiate a SSH rekey. This assurance activity is considered satisfied as the required information has been discovered.

FCS_SSHC_EXT.1.9 – There are no NDcPP AGD assurance activities for this SFR.

FCS_SSHS_EXT.1.1 – There are no NDcPP AGD assurance activities for this SFR.

FCS_SSHS_EXT.1.2 – There are no NDcPP AGD assurance activities for this SFR.

FCS_SSHS_EXT.1.3 – There are no NDcPP AGD assurance activities for this SFR.

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)."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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)."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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)."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. Section 6.5 of the AGD states that the MAC algorithms defined in the ST are the only ones included in the evaluated configuration and that the "none" MAC algorithm is never allowed for SSH. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 6.5 of the AGD states that the SSH session key thresholds for time and amount of transmitted data are not configurable in the evaluated configuration. It also states that whichever threshold (traffic or time) occurs first is when the TOE will initiate a SSH rekey. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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).*

Section 6.9 of the AGD "TLS Functionality" states that hostname reference identifier is the only supported value for the LDAP. Wildcards cannot be defined as part of the reference identifier on the TOE, but the TOE will accept certificates with wildcards in the left-most label (e.g. *.example.com). The TOE supports the SAN extensions for certificate validation. The only Supported Elliptic Curves Extension included in the Client Hello are the NIST curves secp256r1, secp384r1, and secp521r1. This is not configurable. Certificate pinning is not supported. When certificate validation fails, the connection is not established.

Section 7.1.2 of the AGD "LDAP Authentication Configuration" provides detailed instructions on how to configure the TOE. The LDAP configuration assigns the reference identifier from these entries. This assurance activity is considered satisfied as the required information has been discovered.

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.

Section 6.9 states that the hostname reference identifier is the only supported value for the LDAP connection (TLS Client). Therefore, the TOE does not support IP addresses and this assurance activity is considered not applicable.

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."

The TOE is a standalone product and therefore the requirements for a distributed TOE are not applicable.

FCS_TLSC_EXT.1.3 – There are no NDcPP AGD assurance activities for this SFR.

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."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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)."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. Section 6.9 defines the requirements for a TLS client to communicate with the TLS Server. These requirements include:

- TLSv1.2 is the only acceptable version of the TLS protocol and all others will be rejected,
- Only accepts certificates with the wildcard notation in the left-most label,
- supports SAN extension,
- Supported NIST curves: secp256r1, secp384r1, secp521r1
- Neither session resumption nor session tickets are supported.

This assurance activity is considered satisfied as the required information has been discovered.

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

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 6.3 of the AGD describes how to configure the TOE to use Secure Cryptography Mode, which limits the cryptographic options to be consistent with the claims made in the Security Target. This assurance activity is considered satisfied as the required information has been discovered.

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."

Section 7.2.1 of the AGD provides instructions for configuring the number of successive unsuccessful authentication attempts and time period for the length of the lockout.

It also describes that when the lockout period has elapsed, the authentication failure counter is reset to zero and unlocks the account. The authentication failure settings can be configured such that the default 'admin' user is exempt from the authentication failure lockout policy. This assurance activity is considered satisfied as the required information has been discovered.

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."

Section 7.4 of the AGD identifies the set of characters that may be used in passwords and provides suggested guidance to security administrators on the composition of strong passwords. Section 7.4.1 provides the commands to set the minimum password length and the minimum password lengths supported. This assurance activity is considered satisfied as the required information has been discovered.

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_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."*

Section 7.1 of the AGD identifies that the TSF does not echo the user's password while typing, thus masking the password to prevent the password from being shared. This assurance activity is considered satisfied as the required information has been discovered.

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 pre-shared 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."*

Section 7.1 of the AGD describes how to authenticate to the TOE locally using the CLI and remotely using the CLI. Specifically, section 7.1.1 of the AGD describes the steps for configuring the TOE to be able to accept incoming authentication requests from an SSH client using public-key based authentication. Sections 7.1.2 of the AGD describes the steps for configuring the TOE to be able to accept incoming authentication requests via any TOE authentication interface when supplied with LDAP credentials. Section 7.1 of the AGD provides instructions on how to access and authenticate to the TOE via the local console. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 6.6 states that the TOE performs certificate validity checking for outbound TLS connections to the LDAP Server. In addition to the validity checking that is performed by the TOE, the TOE will validate certificate revocation status using a certificate revocation list (CRL) that the TOE is configured to download automatically from a Certification Authority in the Operational Environment. The TOE determines the validity of certificates by ensuring that the certificate and the certificate path are valid. The TOE also ensures that the extendedKeyUsage field includes the correct purpose for its intended use, which includes Server Authentication for TLS server certificates; the TOE does not handle TLS client certificates, certificates associated with OCSP responses, or code signing certificates. In the event that the revocation status cannot be verified, the certificate will not be accepted. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 7.1.2 provides instructions on configuring the TOE to use LDAP Authentication. The instructions include installing the LDAP server certificate on the TOE, configuring the TOE parameters to communicate with the LDAP server, and configuring the aaa authentication parameters on the TOE. Additionally, it identifies CA certificates issued for the LDAP server connection must be ECDSA certificates in order to be used with the ciphersuites claimed as part of this CC evaluation. Section 7.1 states that if the LDAP server is unreachable, the TOE will only perform a single attempt to connect to the LDAP server and will then default to verifying the authentication credentials to the TOE's local store. There is no administrative action to take. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 6.9.1 provides instructions on generating the certificate request message so that its server certificate can be signed by a Certification Authority. Steps 3 & 4 in the instructions provides the commands to establish the *"Common Name", "Organization", "Organizational Unit", and "Country".* This assurance activity is considered satisfied as the required information has been discovered.

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).”

Section 7.8 of the AGD, and its subsections, describes the steps necessary to perform a manual update to the TOE software. 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_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.” The only configuration required is the public-key based configuration, which is covered in section 7.1.1 of the AGD, and LDAP configuration (optional) which is covered in section 7.1.2.

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

- Section 6 covers all required steps to put the TOE into the evaluated configuration
- Sections 6.1, 7.1, and 7.3 cover the requirements to provide the ability to administer the TOE locally and remotely
- Section 7.6 covers the ability to configure the access banner
- Section 7.5.2 covers the ability to configure the session inactivity time before session termination
- Section 7.8 and its subsections cover the ability to update the TOE
- Section 7.2.1 covers the ability to configure the authentication failure parameters for FIA_AFL.1
- Section 6.3 covers the ability to configure the cryptographic functionality

- Sections 6.5.1, 7.1.1, and Section 8.1.1 cover the ability to enable ssh and create host-key and manage cryptographic keys including the public key used for user authentication.
- Section 7.1.2 covers the X509 certificates management for the TLSC connection.
- Section 6.9.1 and 6.9.2 covers certificate management for the TLSS connection for Gigamon FM (separate product)
- Section 7.7 covers the ability to set the time which is used for time-stamps
- Section 7.1.1 covers the ability to manage the trusted public keys database

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."

Section 7.1.1 includes the instructions to generate or load the SSH public/private key pairs for user authentication. Section 8.1.1 includes instruction for generating public key pair for the TOE to log into the audit server. Section 6.5.1 includes instructions for generating the host key for remote connections to the GigaVUE CLI. Section 7.1.2 covers the X509 certificates management for TLSC functionality. Section 6.9.1 and 6.9.2 covers the X509 certificate management for TLSS functionality. This is consistent with the claims made in the ST. Section. 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."

The evaluator created the table below by taking the functions defined the TSS Section 8.4.4 and then mapping the AGD sections to each function.

Management Function	CLI
View Audit Data	Section 8.1
Delete Audit Log	Section 8.1
Configure TLS Connection Parameters	TLSC Section 7.1.2 TLSS Section 6.9.1 and 6.9.2
Configure SSH Connection Parameters	SHHS Section 6.5.1 Section 7.1.1 SHHC: Section 8.1.1
Configure Failed Lockout Threshold	Section 7.2.1
Configure Lockout Duration	Section 7.2.1
Create Users	Section 7.3.1
Modify User Passwords	Section 7.3.2
Modify Password Policy	Section 7.4.1
Configure Supported Authentication Mechanism	Section 7.1.1 Section 7.1.2
Initiate Manual Update	Section 7.8.2
Configure System Time	Section 7.7
Configure Idle Session Timeout	Section 7.5.2
Configure Banner Text	Section 7.6
Manage the Cryptographic Keys	Sections 6.5.1, 7.1.1, and Section 8.1.1
Manage the Trusted Public Keys Database	TLSC Section 7.1.2 TLSS Section 6.9.1 and 6.9.2

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). 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."*

Section 7.1 of the AGD contains instructions for administering the TOE both locally and remotely. Section 6 also provides the configuration requirements for configuring the TOE for remote access for administration and disabling/enabling services. Section 7 includes the instructions on authenticating locally and remotely. This assurance activity is considered satisfied as the required information has been discovered.

FPT_APW_EXT.1 – There are no NDcPP AGD assurance activities for this SFR.

FPT_SKP_EXT.1 – There are no NDcPP AGD assurance activities for this SFR.

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."

Section 7.7 of the AGD describes how the administrator can set the TOE system time via the CLI. The TOE does not obtain time from the underlying VS or NTP server. This assurance activity is considered satisfied as the required information has been discovered.

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."

Section 6.7 of the AGD describes the self-tests in detail and provides examples as to expected outcomes. If the TOE fails any integrity check, cryptography or software integrity, the TOE is put into a safe mode. In safe mode, the device will operate in a limited manner which requires user intervention to bring the appliance back into a normal state after fixing the issues. The console display clearly indicates that the appliance is in safe mode along with the diagnostic information and to contact Gigamon Technical

Support. 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.

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.*

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.

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.

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."

Sections 7.8.1 (CLI) of the AGD describe how to query the currently active TOE software version. Section 7.8.2 states that after the update has been fetched and installed, it resides on a separate partition other than the currently booted partition. The AGD provides instructions on how to query the loaded, but inactive software version.

Section 7.8 of the AGD describes how the verification of the authenticity of the update is performed using a digital signature. It states that all GigaVUEs are pre-loaded with a key for the signature verification performed as part of the update mechanism. Before the actual installation occurs, the signature is verified against the stored key. The image will not be installed if the update fails to be verified. 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.

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."*

Section 7.5.2 of the AGD states that the TOE is designed to terminate a local session after a specified period of time. It also describes the steps on how to configure the CLI timeout period. This assurance activity is considered satisfied as the required information has been discovered.

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

Section 7.5.2 of the AGD states that the TOE is designed to terminate a remote session after a specified period of time. It also describes the steps on how to configure the CLI interface timeout period.
This assurance activity is considered satisfied as the required information has been discovered.

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

Section 7.5.1 of the AGD describes how to terminate both local and remote sessions by executing the "exit" command via the CLI. This assurance activity is considered satisfied as the required information has been discovered.

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

Section 7.6 of the AGD describes how to configure the pre-authentication banner message from the CLI. This assurance activity is considered satisfied as the required information has been discovered.

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."*

Section 8.1.1 of the AGD contains instructions for how to establish a connection to the syslog server using the permitted protocol and Section 8.1 describes the recovery behavior if the connection is interrupted during a log transfer. Section 7.1.2 of the AGD contains instructions for how to configure the TOE to communicate with the LDAP server using the permitted protocol. and describes the behavior if the connection if the LDAP server is unreachable. Section 6.9.2 contains the instruction for how to configure the TOE to communicate with the FM server. This assurance activity is considered satisfied as the required information has been discovered.

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

Section 7.1 of the AGD contains instructions for establishing remote administrative sessions via the CLI using SSH. This assurance activity is considered satisfied as the required information has been discovered.

4 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).

4.1 Platforms Tested and Composition

The evaluation team set up a test environment for the independent functional and vulnerability testing that allowed the team to perform SFR test assurance activities across several of the claimed models and over the relevant interfaces based on the Test Coverage analysis presented in Section 4.2 of this document.

For the HC1, TA200, and GTAP models 100% of the defined tests were executed. This ensured that each defined test was performed on all models and each interface was tested multiple times.

For the HC3, HC1Plus, HCT, TA25, TA25E, and TA200E models, the evaluation team defined a sampling of tests that covers 60% of the defined non-CAVP tests that were used for the fully tested models. The defined sampling was designed to stimulate:

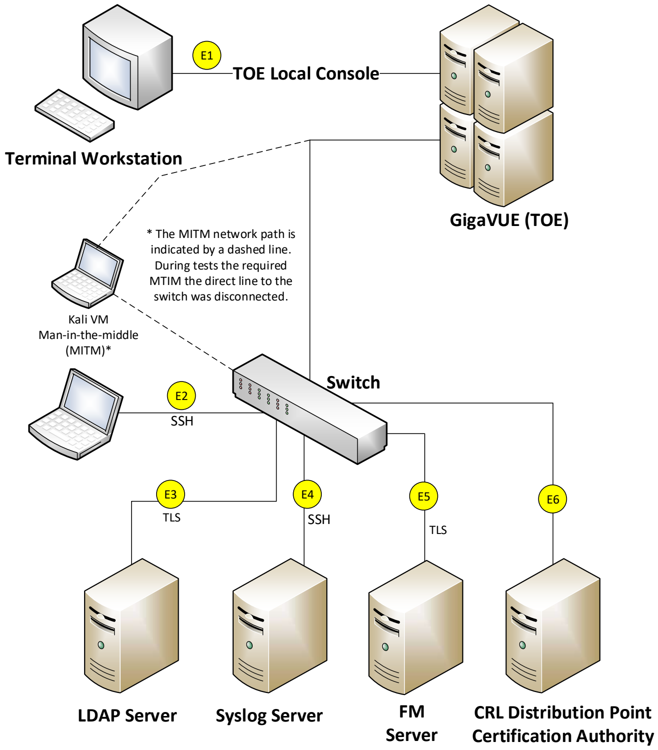
- both local CLI and remote SSH CLI for administrative management
- each method of authenticating to the TOE (password, public key, LDAP)
- each external interface to an external OE entity (LDAP, Syslog, FM Server, CA) when applicable to the model and/or model family (HC, TA, GTAP)
- each secure protocol and client/server functionality (TLSC, TLSS, SSHC, SSHS) when applicable to the model and/or model family (HC, TA, GTAP)

The TA400 will have CAVP verification only.

The justification for assigning a sampling of tests was based on a review of TOE's documentation and functional management capabilities provided by each claimed device. The review revealed that every functional management capability is performed using the same procedures and logical interfaces regardless of model and software image. Additionally, through the course of the testing the evaluation team determined that the actual results for the testing conducted contained no differences between the models tested; further corroborating that the differences in software images does not impact the SFR claims being tested. Therefore, it is sufficient to argue that the actual results for a test performed against the set of models tested would have the same actual results for that particular test, had it been performed on the models not tested and if applicable, any modular configuration that the model was placed in.

4.1.1 Test Configuration

The evaluation team configured the TOE for testing according to the *Gigamon GigaVUE Version 6.5 Supplemental Administrative Guidance for Common Criteria Version 1.0* (AGD) document. The evaluation team set up an isolated test environment for the independent functional testing that allowed them to perform the assurance activities against the TOE over the SFR relevant interfaces. The evaluation team conducted all testing activities of the TOE at the Booz Allen CCTL facility in Laurel, MD between September 2023 and October 2024. Testing was performed against both management interfaces defined in the ST (local CLI and remote CLI).


E1
TOE Local Console
Terminal Workstation
GigaVUE (TOE)
* The MITM network path is indicated by a dashed line. During tests the required MTIM the direct line to the switch was disconnected.
Kali VM
Man-in-the-middle
(MITM)*
E2
SSH
Switch
E3
TLS
E4
SSH
E5
TLS
E6
LDAP Server
Syslog Server
FM Server
CRL Distribution Point Certification Authority

E1: This is the local administrator access to the CLI via a direct connection.
E2: The TOE acts as a SSH server for remote administrator access to the CLI.
E3: The TOE acts as a TLSv1.2 client for accessing an LDAP server interface for authentication services.
E4: The TOE acts as an SSH client for sending audit records to a remote audit server for external audit log storage.
E5: (HC Series and TA Series models only) The TOE acts as a HTTPS (i.e., TLSv1.2) server for connections received from a Gigamon Fabric Manager (separate product) which can be used to provide a central location for the configuration, management, and operation of primary functionality of one or more Gigamon GigaVUE HC and TA Visibility Appliances. The trusted channel interface is considered part of the TOE. The

operational functionality provided by the Gigamon Fabric Manager is not considered part of the TOE.

E6: The TOE interfaces with a Certification Authority (CA) for issuance of server certificates and publication of a Certificate Revocation List (CRL) to determine the validity of certificates presented to the TOE.

4.2 *Omission Justification*

The purpose of this section is to define the Gigamon GigaVUE network device models and their associated interfaces that will be used for testing during the evaluation. This section will thus provide an equivalence argument for the models and interfaces that will not be tested.

Series	**GigaVUE HC Series**				**GigaVUE TA Series**					**GigaTAP A Series**
Component	**HC3**	**HC1**	**HC1Plus**	**HCT**	**TA25**	**TA25E**	**TA200**	**TA200E**	**TA400**	**GTAP**
Model Number	GVS-HC3A1-HW GVS-HC3A2-HW	GVS-HC101-HW GVS-HC102-HW	GVS-HC1P1-HW GVS-HC1P2-HW	GVS-HCT01-HW	GVS-TAX21-HW GVS-TAX22-HW GVS-TAX21A-HW GVS-TAX22A-HW	GVS-TAX21E-HW GVS-TAX22E-HW	GVS-TAC21-HW GVS-TAC22-HW	GVS-TAC21E-HW GVS-TAC22E-HW	GVS-TAC41-HW GVS-TAC42-HW	GTP-ATX21 GTP-ASF21

Table 1: TOE models (21 variations)

4.2.1 Physical Interface Assessment

There are a variable number of physical fixed ports provided by the Gigamon network device equipment. Depending on the model, the number of physical fixed port types can range from none to five different types. The different types of physical fixed ports are: Management Ethernet Port, Serial Console Port, 10/100/1000M Port, 1G/10G Port (QSFP), 1G/10G/25G Port (SFP28), 40/100GB Port (QSFP28), and 400GB/100GB/40GB Port (QSFP-DD/ QSFP28/QSFP). There are also a variable number of additional Configurable Ports that are provided by the optional modular components: port blades/modules, TAP modules, bypass combo modules, and GigaSMART modules.

Through the review of the Security Target and guidance documentation as well as hands on experience with the GigaVUE product, the evaluation team determined that the TOE separates the management traffic from the operational data traffic. The separation of management and operational data is achieved using physically separated ports to provide separate management-plane and a operational-data plane networks.

The evaluation team also determined that all claimed functionality in the Security Target is related to management-plane traffic only. Thus, the independent functional testing only focuses on the ports and their interfaces that are relevant to the management-plane functions (i.e. SFR relevant). This means only the Serial Console and the Management Ethernet Port are in scope as they support:

- the local connection for CLI access (Serial Console Port)
- the remote SSH CLI access (Management Ethernet Port)
- the remote connections to the operational environment's audit (SSH via Management Ethernet Port), LDAP (TLS via Management Ethernet Port), CA server (via Management Ethernet Port), and GigaVUE Fabric Manager (via Management Ethernet Port) (only HC Series and TA Series models)

The ports that support operational data-plane traffic are for GigaVUE's primary functionality which cannot be mapped to any PP functionality. The operational data-plane traffic ports have one of the following purposes:

1. Network Port - Where data arrives at the TOE. The ports which receive copied network data for the TOE. SPAN or TAPs are connected to a network port to provide data into the TOE.
2. Tool Port - Where data leaves the TOE. The ports to which the TOE sends data that has been filtered and directed. Tools are connected to the tool ports and receive copied data from the TOE.

Therefore, the remaining fixed ports: 10/100/1000M, 1G/10G (QSFP), 1G/10G/25G (SFP28), 40/100GB (QSFP28), and 400GB/100GB/40GB (QSFP-DD/ QSFP28/QSFP) ports as well as the ports provided by the configurable ports contain only SFR non-interfering interfaces. For this reason, the independent functional testing of the TOE did not include multiple configurations for models that could support the different optional modular components: port blades, port modules, TAP modules, bypass combo modules, and GigaSMART modules. Thus, none of the independent functional testing stimulated the SFR non-interfering interfaces provided by any of the data-plane traffic ports.

4.2.2 User Interface Assessment

All TOE products support a Command Line Interface (CLI) administrator interface that can be accessed either locally (via Serial Console Port) or remotely (SSH via Management Ethernet Port). There are no other administrative interfaces being claimed.

4.2.3 Interface Assessment Conclusion

Based on the interface assessment, the SFR supporting interfaces of all 21 models should behave in the exact same manner, except for the connection to the Gigamon Fabric Manager. The Gigamon Fabric Manager interface behaves the exact same for all models which support the connection but it is not supported on the GTAP series models.

Interfaces tested – TOE functionality	TOE Series' Group
LDAP – TLS Client (Management Ethernet Port)	HC3, HC1, HC1Plus, HCT, TA25, TA25E, TA200, TA200E, TA400, GTAP
Audit Server – SSH Client (Management Ethernet Port)	HC3, HC1, HC1Plus, HCT, TA25, TA25E, TA200, TA200E, TA400, GTAP
Remote Administrator CLI – SSH Server (Management Ethernet Port)	HC3, HC1, HC1Plus, HCT, TA25, TA25E, TA200, TA200E, TA400, GTAP
CA – Client (Management Ethernet Port)	HC3, HC1, HC1Plus, HCT, TA25, TA25E, TA200, TA200E, TA400, GTAP
Gigamon Fabric Manager – HTTPS Server (Management Ethernet Port)	HC3, HC1, HC1Plus, HCT, TA25, TA25E, TA200, TA200E, TA400
Local CLI (Serial Console Port)	HC3, HC1, HC1Plus, HCT, TA25, TA25E, TA200, TA200E, TA400, GTAP

Table 2: Interface Coverage

4.2.4 Models, Modules and Software Assessment

The following sections assess the models from a hardware and software image perspective.

Series	Model Number	Power Type	Processor	Support Fabric Manager	Differences in Installation Binary	TSF Differences
HC3	GVS-HC3A1-HW	AC power	Intel Atom C2758 (Rangeley)	Yes	Binary 1	No differences with models that support Gigamon Fabric Manager
HC3	GVS-HC3A2-HW	DC power	Intel Atom C2758 (Rangeley)	Yes	Binary 1	No differences with models that support Gigamon Fabric Manager
HC1	GVS-HC101-HW	AC power	Intel Atom **C2538** (Rangeley)	Yes	Binary 2	No differences with models that support Gigamon Fabric Manager
HC1	GVS-HC102-HW	DC power	Intel Atom C2538 (Rangeley)	Yes	Binary 2	No differences with models that support Gigamon Fabric Manager
HC1Plus	GVS-HC1P1-HW	AC power	Intel Atom **C3538** (Denverton)	Yes	Binary 3	No differences with models that support Gigamon Fabric Manager
HC1Plus	GVS-HC1P2-HW	DC power	Intel Atom C3538 (Denverton)	Yes	Binary 3	No differences with models that support Gigamon Fabric Manager
HCT	GVS-HCT01-HW	AC power	Intel Atom C3538 (Denverton)	Yes	Binary 4	No differences with models that support Gigamon Fabric Manager
TA25	GVS-TAX21-HW	AC power	Intel Atom C3538 (Denverton)	Yes	Binary 5	No differences with models that support Gigamon Fabric Manager
TA25	GVS-TAX22-HW	DC power	Intel Atom C3538 (Denverton)	Yes	Binary 5	No differences with models that support Gigamon Fabric Manager
TA25	GVS-TAX21A-HW	AC power	Intel Atom C3538 (Denverton)	Yes	Binary 5	No differences with models that support Gigamon Fabric Manager
TA25	GVS-TAX22A-HW	DC power	Intel Atom C3538 (Denverton)	Yes	Binary 5	No differences with models that support Gigamon Fabric Manager
TA25E	GVS-TAX21E-HW	AC power	Intel Xeon D1518 (Broadwell)	Yes	Binary 6	No differences with models that support Gigamon Fabric Manager
TA25E	GVS-TAX22E-HW	DC power	Intel Xeon D1518 (Broadwell)	Yes	Binary 6	No differences with models that support Gigamon Fabric Manager
TA200	GVS-TAC21-HW	AC power	Intel Xeon **D1527** (Broadwell)	Yes	Binary 7	No differences with models that support Gigamon Fabric Manager
TA200	GVS-TAC22-HW	DC power	Intel Xeon D1527 (Broadwell)	Yes	Binary 7	No differences with models that support Gigamon Fabric Manager
TA200E	GVS-TAC21E-HW	AC power	Intel Xeon D1518 (Broadwell)	Yes	Binary 8	No differences with models that support Gigamon Fabric Manager
TA200E	GVS-TAC22E-HW	DC power	Intel Xeon D1518 (Broadwell)	Yes	Binary 8	No differences with models that support Gigamon Fabric Manager
TA400	GVS-TAC41-HW	AC power	Intel Atom C3538 (Denverton)	Yes	Binary 9	No differences with models that support Gigamon Fabric Manager
TA400	GVS-TAC42-HW	DC power	Intel Atom C3538 (Denverton)	Yes	Binary 9	No differences with models that support Gigamon Fabric

						Manager
GTAP	GTP-ATX21	AC power	Intel Atom C3338 (Denverton)	No	Binary 10	Does not claim requirements (i.e., HTTPS, TLS Server, X509 (3)) or selections (i.e., Audit, Trusted Channel) related to Gigamon Fabric Manager interface. No other functional differences with other models. No differences with other GTAP models.
GTAP	GTP-ASF21	AC power	Intel Atom C3338 (Denverton)	No	Binary 10	Does not claim requirements (i.e., HTTPS, TLS Server, X509 (3)) or selections (i.e., Audit, Trusted Channel) related to Gigamon Fabric Manager interface. No other functional differences with other models. No differences with other GTAP models.

Table 3: Hardware and Software Equivalency Factor Table

4.2.5 Hardware Assessment

For all TOE models, the controller cards and module differences are to provide increased performance and scalability for larger network infrastructures that require these TOE models to copy a larger volume of data-plane traffic and forward it to tools for assessment. These differences are related to the operational intent of the data plan functionality that is unrelated to the SFR functionality claimed by this evaluation.

For all TOE models, the amount of memory (RAM), logical drive capacity, power supplies, number of front and rear bays, main board count, cages, and copper/fiber connectors used in the different models varies. The evaluation team has determined that these differences are unrelated to the SFR functionality claimed by this evaluation and are considered equivalent.

Table 3 above describes all the TOE models claimed within this evaluation. The claimed TOE models were listed as such to provide the most direct comparison between similar models.

The most apparent difference between the models is the supported electrical current. The evaluation team determined that the alternating current (AC) and direct current (DC) models within the table had no difference with the evaluated hardware or software, except for the method used to supply and transform the electricity to power the TOE. Therefore, the logical functions provided by the models should be identical in all security functional relevant areas with their electrical current equivalent within each table. For this reason, the evaluation team has determined that it is sufficient to claim that any testing performed on an AC model will be functionally equivalent and any outcomes derived from that testing would be identical to the DC equivalent model. For these reasons, the evaluation team elected not to test any DC models. Therefore, all DC models are covered by equivalency.

For the TA25 component there are four models available in different pre-configured combinations:

- GVS-TAX21-HW (AC power) all ports enabled
- GVS-TAX22-HW (DC power) all ports enabled
- GVS-TAX21A-HW (AC power) 24 10G/25G ports enabled
- GVS-TAX22A-HW (DC power) 24 10G/25G ports enabled

The two DC models will be covered under equivalency as stated above in this section. The only difference between the two AC models is the number of ports that are enabled. The evaluation team has determined that the number of ports enabled is part of the data plane functionality and that these differences are unrelated to the SFR functionality claimed by this evaluation. Therefore, either the GVS-TAX21-HW or the GVS-TAX21A-HW needs to be tested.

4.2.6 Processor Assessment and Algorithm Certificate Testing Justification

The evaluation team assessed the processors used by the TOE series/models and found that there are three supported processor microarchitectures: Rangeley, Denverton, and Broadwell

The HC1 uses the Intel Atom C2538 processor and the HC3 uses the Intel Atom C2758 processor. The evaluation team determined that the Intel Atom C2538 and the Intel Atom C2758 processors have identical microarchitectures, implement a 64-bit instruction set and support AES New instructions. The HC3 uses the Intel Atom C2758 processor, which compared to the Intel Atom C2538, contains additional cores and threads, cache, TDP, and support for Integrated Intel Quick Assist Technology. These differences affect processor performance and do not affect the TOE security functionality[2]. Since, both processors (C2758 and C2538) implement the same microarchitecture, a 64-bit instruction set, and support AES New instructions they can be considered equivalent.

The TA25E and TA200E use the Intel Xeon D1518 processor and the TA200 uses the Intel Xeon D1527 processor. The evaluation team determined that the Intel Xeon D1518 and the Intel Xeon D1527 processors have identical microarchitectures, implement a 64-bit instruction set and support AES New instructions. The Intel Xeon D1527 processor, which compared to the Intel Xeon D1518, contains Turbo Boost Technology. This difference affects processor performance and does not affect the TOE security functionality[3]. Since, both processors (D1518 and D1527) implement the same

[2] https://ark.intel.com/content/www/us/en/ark/compare.html?productIds=77988,77981

[3] https://ark.intel.com/content/www/us/en/ark/compare.html?productIds=91195,91201

microarchitecture, a 64-bit instruction set, and support AES New instructions they can be considered equivalent.

The GTAP models use the same processor: Intel Atom C3338. Because these two models use the same processor, the two GTAP models can be considered equivalent. The HC1Plus, HCT, TA25, and TA400 use the Intel Atom C3538 processor. The evaluation team determined that the Intel Atom C3338 and Intel Atom C3538 processors have identical microarchitectures, implement a 64-bit instruction set and support AES New instructions. The Intel Atom C3538 processor, which compared to the Intel Atom C3338, contains additional cores and threads, base and turbo frequency capacity, cache, thermal design power (TDP), memory capacity, expansion options, I/O capacity, and support for Integrated Intel Quick Assist Technology[4]. These differences affect processor performance and do not affect the TOE security functionality. Since, both processors (C3338 and C3538) implement the same microarchitecture, a 64-bit instruction set, and support AES New instructions they can be considered equivalent.

Therefore, cryptographic algorithm testing will need to be performed on one model from each bullet:

- One HC1 model using Intel Atom C2538 (Rangeley) or one HC3 model using Intel Atom C2758 (Rangeley)
- One model from TA25E, TA200E using Intel Xeon D1518 (Broadwell) or one TA200 using Intel Xeon D1527 (Broadwell)
- One model from HC1Plus, HCT, TA25, TA400 using Intel Atom C3538 (Denverton) or one model from GTAP using Intel Atom C3338 (Denverton)

4.2.7 Hardware Assessment Conclusion

Based strictly on the hardware assessment:

- HC1 and HC3 series are equivalent.
- TA25E, TA200E and TA200 series are equivalent.
- HC1Plus, HCT, TA25, TA400, GTAP series are equivalent.

To further support any equivalency claim amongst the TOE models, the software must be analyzed further.

[4] https://ark.intel.com/content/www/us/en/ark/compare.html?productIds=97929,97928

4.2.8 Software Assessment

The next difference between the models assessed by the evaluation team is the software binary images supplied for each model. There is a total of ten different software binary images. Table 3 describe the models that have been claimed within this evaluation and their key equivalency attributes, including the different binary images.

Each model series has its own binary image. Gigamon asserts that all management plane software that is mapped to the SFR functionality defined in the NDcPP2.2E is exactly the same in code and functionality across all series, except that the GTAP series models does not support a connection with the Gigamon Fabric Manager. The following non-TSF differences in the models, impact compiling the image for a specific model:

1. The primary difference between many GigaVUE models is their support of different sets of data-plane ports, both fixed ports and ports provided by the optional modular components: port blades, port modules, TAP modules, bypass combo modules, and GigaSMART modules. Thus, only support for the data-plane fixed ports and optional modular components available to that model are included within a model’s software image.

2. There are minor differences in the software images to support the processors. Refer to the Processor Assessment and Algorithm Certificate Testing Justification section for further review of the different processors.

To corroborate Gigamon’s assertions, the evaluation team reviewed the ST and supplemental AGD to ensure that the TSS and administrative procedures did not describe any differences between the models which would lead the evaluation to believe that the different images could impact the SFRs in a manner not described. The evaluators determined the following:

- The evaluation team found that the ST and AGD both made it clear that connections between a TOE and a GigaVUE Fabric Manager applies to HC and TA Series models and not the GTAP Series models. This means that SFRs (i.e., FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1, FIA_X509_EXT.3) and selections (i.e., FAU_GEN.1, FTP_ITC.1) related to this functionality would not be applicable to GTAP models. Note, as this difference has already been noted, it will not be included in other bullets below.
- The evaluation team determined that execution of the POST functions (automatic at startup) under FPT_TST_EXT.1 would result in different checks due to the different software images and differences in hardware, specifically the optional modular components. However, this difference would be expected on any product with multiple images and different hardware components, and thus is a by-product of Gigamon’s asserted differences. Additionally, there is no impact on the evaluation team’s testing since the NDcPP does not define any specific type of

functional testing assurance activities other than verifying the claimed tests are carried out for this requirement.

- The supplemental AGD does not include any steps or procedures that differ among the models. All procedures to place the TOE models into configuration and administrate the TOE are the same.
- The CLI interface functionality (local or remote SSH), that is mapped to the SFR functionality defined in the NDcPP2.2E, behave in the exact same manner for all TOE models. Operational configuration of the data-plane is expected to be different as the intended use of the HC vs TA vs GTAP models is different but this functionality is non-TSF.

4.2.9 Software Assessment Conclusion

Based on the above differences, the evaluation team has determined that none of the differences in software for the HC and TA Series models has an impact on the SFR functionality claimed in the ST, ability to be managed per the AGD, and confirmed through testing. Therefore, equivalency can be considered between the HC and TA models. The GTAP Series models do have a subset of functionality to these other models as they do not support the connection to the Gigamon Fabric Manager. The evaluation team has determined that there are no differences in the GTAP model's software as they use the same software binary image. Therefore, equivalency can be considered between the GTAP models.

4.3 Results of Test Coverage Analysis

Based on the results of the interface, hardware, software, and CPU equivalency assessment, the following models will be fully CC tested:

- one HC1 model due being in one of the three different processor groups, having its own image binary, and non-evaluated operational use.
- one TA200 model due being in one of the three different processor groups, having its own image binary, and non-evaluated operational use.
- one GTAP model due being in one of the three different processor groups, having its own image binary, non-evaluated operational use, and being a subset of functionality provided by the other models (does not support Gigamon Fabric Manager connection).

Comparison of these models will verify that differences in CPU, processors, binary images, and subset of functional claims did not result in differences between the same functionality claimed between these models. Additionally, one model from HC3, HC1Plus, HCT, TA25, TA25E, and TA200E models will be partially tested. The TA400 will only have CAVP verification performed. A sampling from the different SFR families will be conducted. This will further confirm that the claim of ‘equivalent functionality even though it uses a different installation binary’ is accurate.

The results of the evaluation across all models will be reviewed by the evaluation team to ensure that there is equivalence in functionality for all models, per the SFR claims against that model Thus, the following is expected to be demonstrated:

- All HC and TA models will be found fully functionally equivalent as they relate to the evaluation.
- The GTAP models will be found fully functionally equivalent to the HC and TA models for the subset of claimed functionality in the evaluation.

*4.4 **Test Cases***

The evaluation team completed the functional testing activities within the Booz Allen 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]. 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. FCS_SSH_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.

4.4.1 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. Authenticate to the TOE via SSH. 2. Configure logging levels for audit records and cli commands by entering the following commands on the TOE: enable config terminal logging level audit mgmt info logging level cli commands info 3. On the TOE, enter the following commands to turn off local audit logging: logging local none 4. Examine the local and/or remote log repository and verify that audit logs were generated for the shutdown of audit functionality. 5. On the TOE, enter the following commands to turn on local audit logging: logging local info 6. Examine the local and/or remote log repository and verify that audit logs were generated for the startup of audit functionality. 7. Collect audit logs for the other actions defined under this assurance activity while performing other test assurance activities throughout the evaluation. NOTE: Audit Records for the establishment and termination of channels for each cryptographic protocol is performed in other tests. Please see FAU_GEN.1_001.txt for a mapping of the audit records to the tests.
Test Results	Each event in the Security Target that requires an associated audit record was mapped by the evaluator. The evaluator confirmed that all required audit records were generated and contained all the required fields as identified in FAU_GEN.1.2 and FAU_GEN.2.1 - 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	003
SFR	FAU_STG.1
Test Objective	The evaluator shall perform the following tests: Test 1: The evaluator shall access the audit trail without authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all) and attempt to modify and delete the audit records. The evaluator shall verify that these attempts 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 access the audit trail 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. For distributed TOEs the evaluator shall perform test 1 and test 2 for each component that is defined by the TSS to be covered by this SFR.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the CLI as ‘limiteduser’. 2. Execute the following commands: enable config terminal log files delete oldest 3. Verify that the command fails to execute. 4. Attempt to overwrite the TOE local audit file as ‘limiteduser’ by executing the transfer command from a test machine. 5. Verify that no log files are modified or deleted.
Test Results	The evaluator confirmed that the TOE prevented a non-Security Administrator user from deleting and overwriting audit files - Pass
Execution Method	Manual

Test Case Number	004
SFR	FAU_STG.1

Test Objective	The evaluator shall perform the following tests: Test 2: The evaluator shall access the audit trail as an authorized administrator and attempt to delete the audit records. The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records authorized for deletion are deleted. For distributed TOEs the evaluator shall perform test 1 and test 2 for each component that is defined by the TSS to be covered by this SFR.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the CLI as 'admin'. 2. Execute the following commands: enable config terminal log files delete oldest show log files 3. Verify that the command executes successfully and that the specified log file is deleted.
Test Results	The evaluator confirmed the TOE allowed a Security Administrator to delete a specified audit file - 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: 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. Authenticate to the TOE via SSH. 2. Configure the TOE to enable automatic secure transmission of log data to a remote syslog server by entering the following commands: enable config terminal ssh client user admin identity ecdsa generate show ssh client 3. Copy the ECDSA public key that was generated for the admin user in Step 2. 4. On the remote syslog server, insert the public key that was copied in Step 3 into the /home/cctl/.ssh/authorized_keys file on the syslog server. 5. Begin capturing packets on the test machine.

	6. On the TOE, enter the following commands: logging [SYSLOG_SERVER_IP] tcp 514 ssh username cctl logging level audit mgmt info logging level cli commands info logging trap info (a) If a host key verification error is returned, remove the cached SSH host key: ssh client user admin known-host [SYSLOG_SERVER_IP] remove (i) Stop and start the remote audit transmission: logging trap none logging trap info 7. Perform some actions on the TOE that cause audit logs to be generated. 8. Stop capturing packets on the test machine. 9. Examine the captured packets and verify that the data transmitted from the TOE to the remote syslog server are encrypted. 10. Record the remote audit server name and version.
Test Results	The evaluator confirmed that the captured packets showed that a SSHv2 encrypted channel was used to protect the syslog traffic - 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: 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. Authenticate to the TOE via the CLI as the "admin" user. 2. Execute the following commands to increase local logging to the maximum level on the TOE: enable config terminal logging local info

	logging level audit mgmt info logging level cli commands info 3. Execute the following commands on the TOE that causes audit logs to be generated until the messages file reaches 8MB: If "G-TAP", execute: NOTE: Repeat this sequence sufficient number of times with a sufficient number of carriage returns between each repetition. show system ([ALT] + [013]) Otherwise, execute: no chassis box-id 1 chassis box-id 1 4. Repeat Step 3 until the local audit storage space reaches its maximum capacity. 5. Verify that the logs rotated on the TOE such that the data in "messages" is moved to "messages.1.gz", the data from "messages.1.gz" is moved to "messages.2.gz", and so on ending with the data from "messages.7.gz" moved to "messages.8.gz" and the data in "messages.8.gz" deleted.
Test Results	The evaluator confirmed that the TOE rotates the logfiles, as claimed in the Security Target - 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: 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	N/A
Test Steps	N/A - FAU_STG_EXT.2/LocSpace is not claimed in the Security Target.
Test Results	N/A
Execution Method	N/A

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: 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	N/A
Test Steps	N/A – The TOE is not a distributed TOE.
Test Results	N/A
Execution Method	N/A

4.4.2 Cryptographic Support

Test cases for FCS_CKM.1, FCS_CKM.2, 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 CAVP Certificates A4848 and A4849.

SFR(s) Supported	Algorithm(s) (cryptographic operation)	Standard	CAVP Algorithm List Name	CAVP Cert. #
FCS_CKM.1 **Key Generation**	ECDSA (P-256, P-384, P-521)	NIST FIPS 186-4	ECDSA	A4848
FCS_CKM.2 **Key Establishment**	Elliptic curve-based key establishment schemes	NIST SP 800-56A Rev3	KAS-SSC ECC / KAS-ECC CDH	A4849
FCS_COP.1/DataEncryption **AES Encryption /Decryption**	AES-CBC (128, 256 bits) AES-GCM (128, 256 bits)	ISO 10116 (CBC) ISO 19772 (GCM) ISO 18033-3 (AES)	AES	A4848
FCS_COP.1/SigGen **Sig Generation /Verification**	Elliptic Curve Digital Sig Algorithm (256 bits, NIST curve P-256, P-384, P-521)	ISO/IEC 14888-3, Section 6.4. (NIST FIPS 186-4)	ECDSA	A4848
FCS_COP.1/Hash **Cryptographic Hashing**	SHA-256, SHA-384, SHA-512 Digest sizes 256, 384, 512	ISO/IEC 10118-3:2004	SHS	A4848
FCS_COP.1/KeyedHash **Keyed Hash Algorithm**	HMAC-SHA-256, HMAC-384, HMAC-SHA512 Key Sizes 256, 512 bits Digest Sizes 256, 384, 512	ISO/IEC 9797-2:2011, Section 7	HMAC	A4848

FCS_RBG_EXT.1 **Random Bit Generation**	CTR_DRBG (AES-256) with 2 software-based noise sources with minimum of 256 bits of entropy	ISO/IEC 18031:2011	DRBG	A4848

Test Case Number	009
SFR	FCS_CKM.1 - TD0580
Test Objective	Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1.
Test Instructions	N/A
Test Steps	N/A – Per ST, the TOE does not claim FFC Schemes using safe-prime groups
Test Results	N/A
Execution Method	N/A

Test Case Number	010
SFR	FCS_CKM.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	N/A
Test Steps	N/A – Per ST, the TOE does not claim FFC Schemes using safe-prime groups
Test Results	N/A
Execution Method	N/A

Test Case Number	110
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	Testing of this assurance activity is performed with FIA_X509_EXT.1/Rev testing.
Test Results	Pass
Execution Method	Manual

Test Case Number	012
SFR	FCS_SSHC_EXT.1.2 – TD0636
Test Objective	Test objective: The purpose of these tests is to check the authentication of the client to the server using each claimed authentication method. Test 1: For each claimed public-key authentication method, the evaluator shall configure the TOE 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 server to demonstrate the use of all claimed 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 TOE, generate an ecdsa-sha2-nistp384 SSH keypair. 2. On the remote SSH server, configure the SSH server authorized keys using the ecdsa-sha2-nistp384 public key generated from Step 1. 3. Connect to the SSH server from the TOE and confirm that the connection was successful.
Test Results	The evaluator confirmed that the TOE's SSH connection attempt to the SSH server was successfully established using the specified public-key based user authentication algorithm of ecdsa-sha2-nistp384 - Pass
Execution Method	Manual

Test Case Number	108
SFR	FCS_SSHC_EXT.1.2 – TD0636
Test Objective	Test objective: The purpose of these tests is to check the authentication of the client to the server using each claimed authentication method. Test 2: [Conditional] If password-based authentication method has been selected in the FCS_SSHC_EXT.1.2, then following the guidance documentation the evaluator shall configure the TOE to perform password-based authentication with a remote SSH server to demonstrate that the TOE can successfully authenticate using a password as an authentication method.
Test Instructions	N/A
Test Steps	N/A – Password-based authentication is not selected in the ST.
Test Results	N/A
Execution Method	N/A

Test Case Number	013
SFR	FCS_SSHC_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. Authenticate to the TOE via the CLI using SSH. 2. On the TOE, execute the following commands to initiate a connection to the remote SSH server: enable config terminal image fetch scp://[USER]@[TEST_MACHINE_IP_ADDRESS]/home/cctl/bigfile bigfile 3. Verify large packet was dropped
Test Results	The evaluator confirmed that the connection is dropped once a large packet exceeding the ST defined value for this SFR is received - Pass
Execution Method	Manual

Test Case Number	014
SFR	FCS_SSHC_EXT.1.4
Test Objective	The evaluator must ensure that only claimed ciphers and cryptographic primitives are used to establish a SSH connection. To verify this, the evaluator shall start session establishment for a SSH connection with a remote server (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. Authenticate to the TOE via the CLI using SSH. 2. Execute the following commands on the TOE: enable config terminal logging trap none 3. Begin capturing packets between the TOE and the remote audit (SSH) server. 4. Execute the following command on the TOE to initiate a connection to the remote SSH server: logging trap info 5. Perform some activity on the TOE to cause it to transmit audit data to the remote audit (SSH) server. 6. Stop capturing packets between the TOE and the remote audit (SSH) server. 7. Inspect the packet capture and verify that the TOE offers only the ciphers defined in the Security Target and no other ones.
Test Results	The evaluator confirmed that the TOE's SSH client algorithms are consistent with the selections and assignments chosen in the ST for this requirement and all other FCS_SSHC_EXT.1 related requirements. There were no unclaimed algorithms present - Pass
Execution Method	Manual

Test Case Number	015
SFR	FCS_SSHC_EXT.1.5
Test Objective	Test 1: The evaluator shall establish an SSH connection using each of the public key algorithms specified by the requirement to authenticate an SSH server to the TOE. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. Test objective: The purpose of this positive test is to check the authentication of the server by the client (when establishing the transport layer connection), and not for checking generation of the authentication message from the client (in the User Authentication Protocol). The evaluator shall therefore establish sufficient separate SSH connections (with an appropriately configured server) to cause the TOE to demonstrate use of all public key algorithms claimed in FCS_SSHC_EXT.1.5 in the ST.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Configure the remote test SSH server to permit only the ecdsa-sha2-nistp256

	public key algorithm. 2. Authenticate to the TOE via the CLI using SSH. 3. Execute the following commands on the TOE: enable config terminal logging trap none ssh client user admin known-host [SYSLOG_SERVER_IP] remove 4. Begin capturing packets between the TOE and the remote audit (SSH) server. 5. Execute the following command on the TOE to initiate a connection to the remote SSH server: logging trap info 6. Perform some activity on the TOE to cause it to transmit audit data to the remote audit (SSH) server. 7. Stop capturing packets between the TOE and the remote audit (SSH) server. 8. Inspect the packet capture and verify that the TOE offers only the ecdsa-sha2-nistp256 public key algorithm and that the connection was successful. 9. Repeat Steps 1-8, except replace “ecdsa-sha2-nistp256” with “ecdsa-sha2-nistp384”. 10. Repeat Steps 1-8, except replace “ecdsa-sha2-nistp256” with “ecdsa-sha2-nistp521”.
Test Results	The evaluator confirmed that the TOE's SSH connection attempts to the SSH server using each of the TOE's claimed SSH public key algorithms (ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521) were successfully established - Pass
Execution Method	Manual

Test Case Number	016
SFR	FCS_SSHC_EXT.1.5
Test Objective	Test 2: The evaluator shall configure an SSH server to only allow a public key algorithm that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection from the TOE to the SSH server and observe that the connection is rejected.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the CLI using SSH. 2. Execute the following commands on the TOE: enable config terminal logging trap none 3. Begin capturing packets between the TOE and the remote audit (SSH) server. 4. Execute the following command on the TOE to initiate a connection to the remote SSH server: logging trap info 5. Perform some activity on the TOE to cause it to transmit audit data to the remote audit (SSH) server. 6. Stop capturing packets between the TOE and the remote audit (SSH) server. 7. Inspect the packet capture and verify that the connection was unsuccessful.

Test Results	The evaluator confirmed that the TOE's SSH connection attempts to the SSH server configured to use the disallowed ssh-dss algorithm failed - Pass
Execution Method	Manual

Test Case Number	017
SFR	FCS_SSHC_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	1. Configure the remote SSH server to only allow the hmac-sha2-256 integrity algorithm. 2. Authenticate to the TOE via the CLI using SSH. 3. Execute the following commands on the TOE: enable config terminal logging trap none 4. Begin capturing packets between the TOE and the remote audit (SSH) server. 5. Execute the following command on the TOE to initiate a connection to the remote SSH server: logging trap info 6. Perform some activity on the TOE to cause it to transmit audit data to the remote audit (SSH) server. 7. Stop capturing packets between the TOE and the remote audit (SSH) server. 8. Inspect the packet capture and verify that the hmac-sha1 integrity algorithm was used to negotiate the connection. 9. Additionally, inspect the packet capture and verify that the TOE offers only the hmac-sha1, hmac-sha2-256, and hmac-sha2-512 integrity algorithms and that the connection was successful. 10. Repeat Steps 1-9, except in Steps 1 and 8 replace hmac-sha2-256 with hmac-sha2-512.
Test Results	The evaluator confirmed that TOE's SSH connections to the SSH server using each of the TOE's claimed SSH HMAC algorithms (hmac-sha2-256, hmac-sha2-512) were successfully established - Pass
Execution Method	Manual

Test Case Number	018
SFR	FCS_SSHC_EXT.1.6
Test Objective	Test 2: [conditional, if an HMAC or AEAD_AES_*_GCM algorithm is selected in the ST] The evaluator shall configure an SSH server to only allow a MAC algorithm that is not included in the ST selection. The evaluator shall attempt to connect from the TOE to the SSH server 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	1. Authenticate to the TOE via the CLI using SSH. 2. Execute the following commands on the TOE: enable config terminal logging trap none 3. Begin capturing packets between the TOE and the remote audit (SSH) server. 4. Execute the following command on the TOE to initiate a connection to the remote SSH server: logging trap info 5. Perform some activity on the TOE to cause it to transmit audit data to the remote audit (SSH) server. 6. Stop capturing packets between the TOE and the remote audit (SSH) server. 7. Inspect the packet capture and verify that the connection was unsuccessful.
Test Results	The evaluator confirmed that the TOE's SSH connection attempts to the SSH server configured to use the disallowed hmac-md5 algorithm failed. Audit records were properly generated for the failed connection - Pass
Execution Method	Manual

Test Case Number	019
SFR	FCS_SSHC_EXT.1.7
Test Objective	Test 1: The evaluator shall configure an SSH server to permit all allowed key exchange methods. The evaluator shall attempt to connect from the TOE to the SSH server using each allowed key exchange method, and observe that each attempt succeeds.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the CLI using SSH. 2. Execute the following commands on the TOE: enable config terminal 3. Execute the following command to access the TOE debug shell: debug shell req 4. Provide the challenge phrase to the vendor. 5. Execute the following command to provide the challenge response from the vendor to access the debug shell: debug shell enter <response code> 6. Modify the “KexAlgorithms” line in “/etc/ssh/ssh_config” to the following: KexAlgorithms "ecdh-sha2-nistp256"

	7. Authenticate to the TOE in a new session via the CLI using SSH. 8. Execute the following commands on the TOE: enable config terminal logging trap none 9. Begin capturing packets between the TOE and the remote audit (SSH) server. 10. Perform some activity on the TOE to cause it to transmit audit data to the remote audit (SSH) server: logging trap info 11. Stop capturing packets between the TOE and the remote audit (SSH) server. 12. Inspect the packet capture and verify that the ecdh-sha2-nistp256 key exchange method was used to negotiate the connection. 13. Repeat Step 6 except replace "ecdh-sha2-nistp256" with "ecdh-sha2-nistp384". 14. Repeat Steps 8 through 12 except replace "ecdh-sha2-nistp256" with "ecdh-sha2-nistp384". 15. Repeat Step 6 except replace "ecdh-sha2-nistp256" with "ecdh-sha2-nistp521". 16. Repeat Steps 8 through 12 except replace "ecdh-sha2-nistp256" with "ecdh-sha2-nistp521".
Test Results	The evaluator confirmed that TOE's SSH connections to the SSH server using each of the TOE's claimed key exchange algorithms (ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521) were successfully established - Pass
Execution Method	Manual

Test Case Number	020
SFR	FCS_SSHC_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 the TOE to connect to an SSH server 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 server the TOE is connected to. For testing of the traffic-based threshold the evaluator shall use the TOE to connect to an SSH server 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_SSHC_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 server the TOE is connected to. 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 (60 minutes maximum):** 1. Initialize a test SSH server with the following configuration to ensure that the test SSH server does not perform a rekey before the TOE. RekeyLimit=10G 10h 2. Authenticate to the TOE via the CLI using SSH. 3. On the TOE, execute the following commands to initiate a connection to the remote SSH server: logging trap none logging trap info 4. After 30 minutes has elapsed, inspect the audit logs and verify there is a SSH rekey event. **b) Traffic-based Rekey (1GB maximum):** 1. Initialize a test SSH server with the following configuration to ensure that the test SSH server does not perform a rekey before the TOE. RekeyLimit=10G 10h 2. Authenticate to the TOE via the CLI using SSH. 3. On the TOE, execute the following commands to initiate a connection to the remote SSH server:

	enable config terminal image fetch scp://[USER]@[TEST_MACHINE_IP_ADDRESS]/home/cctl/256MBfile 256MBfile 4. Inspect the audit logs and verify there is a SSH rekey event.
Test Results	The evaluator confirmed that the TOE's SSH Client successfully executed a time based SSH rekey in 60 minutes or less. The evaluator also confirmed that the TOE's SSH Client successfully executed a traffic based SSH rekey in 1 GB or less of exchanged data - Pass
Execution Method	Manual

Test Case Number	021
SFR	FCS_SSHC_EXT.1.9
Test Objective	Test 1: The evaluator shall delete all entries in the TOE's list of recognized SSH server host keys and, if selected, all entries in the TOE's list of trusted certification authorities. The evaluator shall initiate a connection from the TOE to an SSH server. The evaluator shall ensure that the TOE either rejects the connection or displays the SSH server's public key (either the key bytes themselves or a hash of the key using any allowed hash algorithm) and prompts the user to accept or deny the key before continuing the connection.
Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the CLI as the "admin" user. 2. Execute the following commands: enable config terminal logging trap none ssh client global host-key-check ask ssh client user admin known-host [SYSLOG_SERVER_IP] remove logging trap info 3. Verify that the TOE displays the SSH server's public key and prompts to accept or deny the key before continuing the connection.
Test Results	The evaluator confirmed that the TOE prompts the admin user to accept/reject the SSH server's public key when no other public key for that SSH server has been installed on the TOE - Pass
Execution Method	Manual

Test Case Number	022
SFR	FCS_SSHC_EXT.1.9
Test Objective	Test 2: The evaluator shall add an entry associating a host name with a public key into the TOE's local database. The evaluator shall replace, on the corresponding SSH server, the server's host key with a different host key. If 'password-based' is selected for the TOE in FCS_SSHC_EXT.1.2, the evaluator shall initiate a connection from the TOE to the SSH server using password-based authentication, shall ensure that the TOE rejects the connection, and shall ensure that the password was not transmitted to the SSH server (for example, by instrumenting the SSH server with a debugging capability to output received passwords). If 'password-based' is not selected for the TOE in FCS_SSHC_EXT.1.2, the evaluator shall initiate a connection from the TOE to the SSH server using public key-based authentication, and shall ensure that the TOE rejects the connection.

Test Instructions	Execute this test per the test steps.
Test Steps	1. Authenticate to the TOE via the CLI. 2. Execute the following commands on the TOE to remove all host key entries from the TOE SSH client: enable config terminal logging trap none ssh client global host-key-check ask ssh client user admin known-host [SYSLOG_SERVER_IP] remove 3. Execute the following commands on the TOE to initiate a connection to the remote SSH server, thus causing the TOE to associate a host name with a public key into the TOE's local database: logging trap info logging trap none 4. On the remote SSH server, replace the server's host key with a different host key. 5. Begin capturing packets between the SSH client and the TOE. 6. Execute the following command on the TOE to initiate a connection to the remote SSH server: logging trap info 7. Stop capturing packets and verify that the TOE rejects the connection.
Test Results	The evaluator confirmed that the TOE rejects the connection when a SSH server presents a different host key than what has been configured on the TOE. Audit records were properly generated for the failed connection - Pass
Execution Method	Manual

Test Case Number	023
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-nistp256 public key algorithm. 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. 5. Repeat Steps 1 – 4, except in Step 1 replace "ecdsa-sha2-nistp384" with "ecdsa-sha2-nistp384". 6. Repeat Steps 1 – 4, except in Step 1 replace "ecdsa-sha2-nistp384" with

	“ecdsa-sha2-nistp521”.
Test Results	The evaluator confirmed that SSH connection attempts to the TOE were successful when valid SSH public-key based user authentication credentials using either ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, or ecdsa-sha2-nistp521 were supplied - Pass
Execution Method	Manual

Test Case Number	024
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. 2. Using the private key from the keypair generated in Step 1, attempt to authenticate to the TOE via the CLI using SSH with a valid username. 3. Verify that the authentication attempt to the TOE fails.
Test Results	The evaluator confirmed that SSH connection attempts to the TOE were unsuccessful when invalid SSH public-key based user authentication credentials were supplied to the TOE - Pass
Execution Method	Manual

Test Case Number	106
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 correct password authenticator. 2. Verify that the authentication attempt was successful.
Test Results	The evaluator confirmed that SSH connection attempts to the TOE were successful when valid SSH authentication credentials were supplied to the TOE - Pass
Execution Method	Manual

Test Case Number	107
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 an incorrect password authenticator. 2. Verify that the authentication attempt was unsuccessful.
Test Results	The evaluator confirmed that SSH connection attempts to the TOE were unsuccessful when invalid SSH authentication credentials were supplied to the TOE - Pass
Execution Method	Manual

Test Case Number	025
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. Begin capturing packets between the SSH client and the TOE. 2. On the test machine, execute the following command: /opt/CATL-65536/bin/scp 1.5bigfile admin@[TOE_IP_ADDRESS]: 3. Stop capturing packets 4. Verify large packet was dropped
Test Results	The evaluator observed that the TOE drops the packet once a large packet exceeding the ST defined value for this SFR is received - Pass
Execution Method	Manual

Test Case Number	026
SFR	FCS_SSHS_EXT.1.4
Test Objective	The evaluator must ensure that only claimed ciphers and cryptographic primitives are used to establish a SSH connection. To verify this, the evaluator shall start session establishment for a 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 via the CLI using SSH. 3. Stop capturing packets between the SSH client test machine and the TOE. 4. Examine the packet capture to verify that either the aes128-cbc, aes256-cbc, aes128-gcm@openssh.com, or aes256-gcm@openssh.com encryption algorithm is used to negotiate the SSH connection. 5. Additionally, examine the “Server: Key Exchange Init” packet to verify that no other encryption algorithms other than those claimed in the Security Target are in the “encryption_algorithms_server_to_client” string. 6. Terminate the SSH connection.
Test Results	The evaluator confirmed that the TOE's SSH server algorithms are consistent with

	the selections and assignments chosen in the ST for this requirement and all other FCS_SSHS_EXT.1 related requirements. There were no unclaimed algorithms present - Pass
Execution Method	Manual

Test Case Number	027
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 evaluator confirmed that the TOE's SSH server public key algorithm used is ecdsa-sha2-nistp384 - Pass
Execution Method	Manual

Test Case Number	028
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 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 confirmed that the TOE's SSH server rejects authentication attempts when a SSH client presents a public-key without the TOE being configuring to recognize that public-key for authentication - Pass
Execution Method	Manual

Test Case Number	030
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	1. On the test machine, configure the SSH client use only the hmac-sha1 integrity algorithm. 2. Begin capturing packets between the SSH client test machine and the TOE. 3. Authenticate to the TOE via the CLI using SSH. 4. Stop capturing packets between the SSH client test machine and the TOE. 5. Examine the packets to verify that the hmac-sha1 integrity algorithm was used. 6. Terminate the SSH connection. 7. Repeat Steps 1-6 except replace “hmac-sha1” with “hmac-sha2-256.” 8. Repeat Steps 1-6 except replace “hmac-sha1” with “hmac-sha2-512.”
Test Results	The evaluator confirmed that TOE's SSH server can successfully establish a connection using each of the TOE's claimed SSH HMAC algorithms (hmac-sha2-256, hmac-sha2-512) - Pass
Execution Method	Manual

Test Case Number	031
SFR	FCS_SSHS_EXT.1.6
Test Objective	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	1. On the test machine, configure the SSH client to only use the hmac-md5 MAC algorithm. 2. Begin capturing packets between the SSH client test machine and the TOE. 3. Authenticate to the TOE via the CLI using the SSH client. 4. Stop capturing packets between the SSH client test machine and the TOE. 5. Verify that the SSH connection failed to establish.
Test Results	The evaluator confirmed that a connection request to the TOE's SSH server from a SSH client configured to use the disallowed hmac-md5 algorithm failed - Pass
Execution Method	Manual

Test Case Number	032

SFR	FCS_SSHS_EXT.1.7
Test Objective	Test 1: The evaluator shall configure an SSH client to only allow the diffie-hellman-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 using the SSH client. 4. Stop capturing packets between the SSH client test machine and the TOE. 5. Using Wireshark, examine the packet capture log for the SSH "Key Exchange Init" packet sent from the test machine to the TOE. 6. Expand "SSH Protocol" > "SSH Version 2" > "Key Exchange" > "Algorithms" and examine the value under the "kex_algorithms" string to verify diffie-hellman-group1-sha1 was offered by the test machine (client). 7. Verify that the SSH connection failed to establish.
Test Results	The evaluator confirmed that a connection request to the TOE's SSH server from a SSH client configured to use the disallowed diffie-hellman-group1-sha1 algorithm failed for this reason - Pass
Execution Method	Manual

Test Case Number	033
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 ecdh-sha2-nistp256 key exchange algorithm. 2. Begin capturing packets between the SSH client test machine and the TOE. 3. Authenticate to the TOE via the CLI using the SSH client. 4. Stop capturing packets between the SSH client test machine and the TOE. 5. Using Wireshark, examine the packet capture log for the SSH "Key Exchange Init" packet sent to the TOE from the test machine. 6. Expand "SSH Protocol" > "SSH Version 2" > "Key Exchange" > "Algorithms" and examine the value under the "kex_algorithms" string to verify ecdh-sha2-nistp256 was used. 7. Repeat Steps 1-6, except in Steps 1 and 6 replace "ecdh-sha2-nistp256" with "ecdh-sha2-nistp384". 8. Repeat Steps 1-6, except in Steps 1 and 6 replace "ecdh-sha2-nistp256" with "ecdh-sha2-nistp521".
Test Results	The evaluator confirmed that a connection request to the TOE's SSH server from a SSH Client configured to use each of the claimed key exchange algorithms (ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521) were successfully established - Pass
Execution Method	Manual

Test Case Number	034
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 maximum):** 1. Authenticate to the TOE via the CLI 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. Configure the inactivity timeout period for the current session to a value greater than 1 hour (e.g. 90 minutes) by executing the following commands: enable config terminal cli session auto-logout 90 3. Wait 1 hour and verify that the TOE generates an audit record for the SSH rekey performed by the TOE. **b) Traffic-based Rekey (1 GB maximum):** 1. Transfer a 1 GB 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" 1GiBfile admin@[TOE_IP_ADDRESS]: 2. Verify that the TOE generates an audit record for the SSH rekey performed by the TOE.
Test Results	The evaluator confirmed that the TOE's SSH server successfully executed a time based SSH rekey in 60 minutes or less. The evaluator also confirmed that the TOE's SSH server successfully executed a traffic based SSH rekey in 1 GB or less of exchanged data. Audit records were generated with the correct reason for rekey initiation - Pass
Execution Method	Manual

Test Case Number	035
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_128_CBC_SHA256 2. Begin capturing packets between the TOE and the remote server. 3. Cause the TOE to establish a TLS connection to the remote server Authenticate to the TOE via SSH with an LDAP account: testUser1@[IP-ADDRESS]

	4. Stop capturing packets between the TOE and the remote server. 5. Inspect the packet capture and verify that the Server Hello message contains the ciphersuite selected in Step 1. 6. Repeat Steps 1-5, except in Step 1 specify the “TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384” ciphersuite. 7. Repeat Steps 1-5, except in Step 1 specify the “TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256” ciphersuite. 8. Repeat Steps 1-5, except in Step 1 specify the “TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384” ciphersuite.
Test Results	The evaluator confirmed that each of the claimed TLS client ciphersuites were successfully used to connect to the remote server - Pass
Execution Method	Manual

Test Case Number	036
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 confirmed that the TLS connection to the remote server was successful when the server presented a server certificate with the Server Authentication purpose and the TLS connection to the remote server was unsuccessful when the server presented a certificate lacking the Server Authentication purpose - Pass
Execution Method	Manual

Test Case Number	037
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 the remote server, load the RSA certificate and select the

	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ciphersuite. 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 failed to establish a connection to the remote server after receiving the server’s Certificate handshake message.
Test Results	The evaluator confirmed that the TOE disconnects after receiving the server's Certificate handshake message that contained a RSA server certificate while using an ECDSA ciphersuite - Pass
Execution Method	Manual

Test Case Number	038
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. Configure the remote server to use the TLS_NULL_WITH_NULL_NULL ciphersuite. 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 denies the connection to the remote server. b) 1. Open Wireshark and begin capturing packets between the TOE and the TLS server. 2. Run the modification test tool on the test system. 3. Initiate a connection from the TOE to the server such that the modification test tool modifies the appropriate packet. 4. Stop capturing packets with Wireshark. 5. Verify that the client rejects the connection after receiving the Server Hello. c) 1. Open Wireshark and begin capturing packets between the TOE and the

	TLS server. 2. Run the modification test tool on the test system. 3. Initiate a connection from the TOE to the server such that the modification test tool modifies the appropriate packet. 4. Stop capturing packets with Wireshark. 5. Verify that the TOE disconnects after receiving the server’s Key Exchange handshake message.
Test Results	The evaluator confirmed that: a) the TOE denies the connection when the server is configured to use the TLS_NULL_WITH_NULL_NULL ciphersuite. b) the TOE denies the connection after receiving the Server Hello that selects a ciphersuite not presented by the TOE Client Hello message. c) the TOE denies the connection after receiving the server's Key Exchange handshake message with the request to perform a key exchange using an unsupported curve/group - Pass
Execution Method	Manual

Test Case Number	039
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. Open Wireshark and begin capturing packets between the TOE and the TLS server. 2. Run the modification test tool on the test system. 3. Initiate a connection from the TOE to the server such that the modification test tool modifies the appropriate packet. 4. Stop capturing packets with Wireshark. 5. Verify that the client rejects the connection. b) 1. Open Wireshark and begin capturing packets between the TOE and the TLS server. 2. Run the modification test tool on the test system. 3. Initiate a connection from the TOE to the server such that the modification test tool modifies the appropriate packet. 4. Stop capturing packets with Wireshark. 5. Verify that the handshake does not finish successfully, and no application data flows.
Test Results	The evaluator confirmed that: a) the TOE rejects the connection when the TLS version selected by the server in the Server Hello was set to a non-supported TLS version.

	b) the TOE denies the connection when a modification was made to the signature block in the Server's Key Exchange handshake message, the handshake did not finish successfully, and that no application data flowed - Pass
Execution Method	Manual

Test Case Number	040
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. Open Wireshark and begin capturing packets between the TOE and the TLS server. 2. Run the modification test tool on the test system. 3. Initiate a connection from the TOE to the server such that the modification test tool modifies the appropriate packet. 4. Stop capturing packets with Wireshark. 5. verify that the handshake does not finish successfully and no application data flows. b) 1. Open Wireshark and begin capturing packets between the TOE and the TLS server. 2. Run the modification test tool on the test system. 3. Initiate a connection from the TOE to the server such that the modification test tool modifies the appropriate packet. 4. Stop capturing packets with Wireshark. 5. verify that the handshake does not finish successfully and no application data flows. c) 1. Open Wireshark and begin capturing packets between the TOE and the TLS server. 2. Run the modification test tool on the test system. 3. Initiate a connection from the TOE to the server such that the modification test tool modifies the appropriate packet. 4. Stop capturing packets with Wireshark. 5. 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 Results	The evaluator confirmed that: a) the TOE denies the connection when a byte in the Server Finished handshake message is modified (new value: 0x41), the handshake does not finish successfully, and no application data flowed. b) the TOE denies the connection when a garbled message is sent (new value: 0x17) from the server after the server has issued the ChangeCipherSpec message, the handshake does not finish successfully, and no application data flows. c) the TOE denies the connection when one byte in the server's nonce in the Server Hello handshake message is modified (new value: 0x41), and rejects the Server Key Exchange handshake message - Pass
Execution Method	Manual

Test Case Number	041
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: 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 confirmed that the TOE denies the connection when the remote server presents a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension - Pass
Execution Method	Manual

Test Case Number	042
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: 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 confirmed that the TOE denies the connection when the remote server presents 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 - Pass
Execution Method	Manual

Test Case Number	043
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: **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 confirmed that the TOE successfully establishes the connection when the remote server presents a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension - Pass
Execution Method	Manual

Test Case Number	044
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: **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 confirmed that the TOE successfully establishes the connection when the remote server presents a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches - Pass
Execution Method	Manual

Test Case Number	045
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: 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	1. Install a certificate on the server containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.catl.local) and specify the reference identifier of the host to be foo.ldap.catl.local. 2. Begin capturing packets between the TOE and the server. 3. Connect the TOE to the server (e.g. foo.ldap.catl.local). 4. Stop capturing packets between the TOE and the server with Wireshark. 5. Verify the connection fails. 6. Install a certificate on the server containing a wildcard in the left-most label (e.g. *.catl.local), and specify the reference identifier of the host to be with a single left-most label (e.g. ldap.catl.local). 7. Using Wireshark, begin capturing packets between the TOE and the server. 8. Connect the TOE to the server. 9. Stop capturing packets between the TOE and the server. 10. Verify the connection succeeds. 11. Repeat Steps 6-9, except in Step 6, configure the reference identifier of the host to catl.local. 12. Verify that the connection fails. 13. Repeat Steps 6-9, except in Step 6, configure the reference identifier of the host to foo.ldap.catl.local.

	14. Verify that the connection fails.
Test Results	The evaluator confirmed that out of every combination tested, the TOE rejected the connection to the remote server, with the exception being when the server presents a certificate containing a wildcard in the left-most label (e.g. *.catl.local), and the reference identifier of the host is specified in the following format: (e.g. <remote-peer>.catl.local), which is the expected behavior - Pass
Execution Method	Manual

Test Case Number	046
SFR	FCS_TLSC_EXT.1.2 – TD0634
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: Objective: The objective of this test is to ensure the TOE is able to differentiate between IP address identifiers that are not allowed to contain wildcards and other types of identifiers that may contain wildcards. Test 6: [conditional] If IP address identifiers 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... This negative test corresponds to the following section of the Application Note 64/105: "The exception being, the use of wildcards is not supported when using IP address as the reference identifier."
Test Instructions	N/A
Test Steps	N/A – The TOE does not support the use of IP address reference identifiers. Therefore, this conditional test does not apply.
Test Results	N/A
Execution Method	N/A

Test Case Number	047
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: 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	N/A
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	N/A
Execution Method	N/A

Test Case Number	048
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. 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. 4. Verify connection succeeds
Test Results	The evaluator confirmed that the TOE's connection to the remote peer was successful when the root CA certificate that is needed to validate the presented certificate was installed on the TOE - Pass
Execution Method	Manual

Test Case Number	049
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. 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. 4. Verify connection fails
Test Results	The evaluator confirmed that the TOE denies the connection when the intermediate 01 CA certificate was removed from the server presented certificate chain. The ST selected "Not implement any administrator override mechanism"; therefore, no additional testing was performed for this assurance activity - Pass
Execution Method	Manual

Test Case Number	050
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	N/A
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	N/A
Execution Method	N/A

Test Case Number	051
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 secp256r1 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. 6. Repeat Steps 1-5, except in Step 1, replace “secp256r1” with “secp384r1”. 7. Repeat Steps 1-5, except in Step 1, replace “secp256r1” with “secp521r1”.
Test Results	The evaluator confirmed that the TOE's connection to the remote peer was successful when using each of the claimed elliptic curves (secp256r1, secp384r1 and secp521r1 - Pass
Execution Method	Manual

Test Case Number	111
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	**Gigamon FM Machine (Client) to GigaVUE Machine (Server)** 1. The following ciphersuites are configured for use by the Client: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 2. Using Wireshark, begin capturing packets between the TOE and the test machine. 3. Connect to the Command Center via the remote workstation web browser. 4. Stop capturing packets with Wireshark. 5. Verify the connection succeeded.
Test Results	The evaluator confirmed that the TOE successfully established the connection for each of the ciphers declared in the Security Target as expected - Pass
Execution Method	Manual

Test Case Number	112
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	**Gigamon FM Machine (Client) to GigaVUE Machine (Server)** (a) Unsupported ciphersuites:

	1. Begin capturing packets between the Workstation and the Command Center. 2. Configure the Command Center to use the following list of ciphersuites: openssl s_client -connect <ip address>:443 -tls1_2 -cipher ECDHE-ECDSA-AES128-SHA 3. Initiate a connection between the Command Center from the Workstation. 4. Stop capturing packets. 5. Verify that the TLS connection could not be established. (b) TLS_NULL_WITH_NULL_NULL: 1. Begin capturing packets between the workstation and the TLS client. 2. Run Ettercap using the Ettercap Filter generated in Setup on the MITM test system by executing the following command: ettercap -Tq -i eth0 -B eth1 -F <filter> 3. Initiate a connection between the TOE from the Console. 4. Stop capturing packets. Verify that the TLS connection could not be established and the server refused to negotiate a ciphersuite.
Test Results	The evaluator confirmed that the TOE correctly failed to establish the connection for ciphers not declared in the Security Target as expected - Pass
Execution Method	Manual

Test Case Number	113
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	**Gigamon FM Machine (Client) to GigaVUE Machine (Server)** a) 1. Begin capturing packets between the TOE and the TLS client. 2. Run Ettercap using the Ettercap filter generated in Setup on the MITM test system by executing the following command: ettercap -Tq -i eth0 -B eth1 -F <filter> 3. Initiate a connection from the TLS client to the TOE such that Ettercap modifies the appropriate packet. 4. Stop capturing packets. 5. Confirm the TLS connection failed to establish. b) 1. Open Wireshark and begin capturing packets between the TOE and the TLS client. 2. Initiate a connection from the TLS client to the TOE. 3. Stop capturing packets. 4. 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 confirmed that the TOE correctly rejects/denies the modified traffic and properly establishes the non-modified traffic. The Finished message was sent immediately after the server's ChangeCipherSpec message and did not contain unencrypted data as expected - Pass
Execution Method	Manual

Test Case Number	114

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	**Gigamon FM Machine (Client) to GigaVUE Machine (Server)** 1. Begin capturing packets between the Client and Server. 2. Execute the following commands on the Client to initiate a connection to the TOE using the disallowed protocols: openssl s_client -connect <TOE_IP_ADDRESS>:443 -tls1_1 openssl s_client -connect <TOE_IP_ADDRESS>:443 -tls1 openssl s_client -connect <TOE_IP_ADDRESS>:443 -ssl2 openssl s_client -connect <TOE_IP_ADDRESS>:443 -ssl3 3. Stop capturing packets and verify that the connection(s) failed for the unsupported protocol versions in the SFR. 4. Begin capturing packets between the Client and Server. 5. Execute the following commands on the Client to initiate a connection to the TOE using the disallowed protocols: openssl s_client -connect <TOE_IP_ADDRESS>:443 -tls1_2 6. Stop capturing packets and verify that the connection(s) succeeds for the supported protocol versions in the SFR.
Test Results	The evaluator confirmed that the TOE correctly failed to establish the connection for the TLS client requested to use protocol versions TLSv1.0, TLSv1.1, SSLv2.0, and SSLv3.0. The evaluator observed that the TOE correctly established the connection when the TLS client requested to use protocol version TLSv1.2 - Pass
Execution Method	Manual

Test Case Number	115
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. 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	**Gigamon FM Machine (Client) to GigaVUE Machine (Server)** 1. Configure the Command Center to use the secp521r1 elliptic curve. 2. Begin capturing packets between the TOE and the Command Center. 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. 6. Repeat steps 1-5 except replace secp521r1 with secp384r1. 7. Repeat steps 1-5 except replace secp521r1 with secp256r1
Test Results	The evaluator confirmed that the TOE correctly established the connection with all elliptic curves declared in Security Target - Pass
Execution Method	Manual

Test Case Number	116
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	N/A
Test Steps	N/A – ECDHE is the only key establishment supported.
Test Results	N/A
Execution Method	N/A

Test Case Number	117
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). 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) 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) 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. 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 Instructions	Execute this test per the test steps.
Test Steps	**Gigamon FM Machine (Client) to GigaVUE Machine (Server)** 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 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; otherwise perform the following steps: a. Capture the SessionID from the Server Hello. b. Send a new Client Hello containing the captured Session ID. 6. Verify that the TOE rejects the SessionID by sending a Server Hello with a different SessionID and by performing a full handshake.
Test Results	The evaluator confirmed that the Client Hello is sent with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket. There is no presence of a NewSessionTicket handshake message (at any point in the handshake). The Server Hello message contains a zero-length session identifier - Pass
Execution Method	Manual

Test Case Number	118
SFR	FCS_TLSS_EXT.1.4 - TD0569
Test Objective	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. Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ID may be obtained in one context for resumption in another context. There is no requirement that the session ID be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ID constructed in another context must be tested. 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 Instructions	N/A
Test Steps	N/A – The Security Target does not specify that the TOE supports session resumption using session IDs; therefore, this conditional test does not apply.
Test Results	N/A
Execution Method	N/A

Test Case Number	119
SFR	FCS_TLSS_EXT.1.4 – TD0555, TD0556, TD0569
Test Objective	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. Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ticket may be obtained in one context for resumption in another context. There is no requirement that the session ticket be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ticket constructed in another context must be tested. 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 Instructions	N/A
Test Steps	N/A – The Security Target does not specify that the TOE supports session resumption using session tickets; therefore, this conditional test does not apply.
Test Results	N/A
Execution Method	N/A

4.4.3 Identification and Authentication

Test Case Number	062
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): 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):** 1. Authenticate to the TOE via the CLI. 2. Enter the following commands: enable config terminal 3. Enter the following commands to configure the number of successive unsuccessful authentication attempts before the account is locked and the time period that it remains locked. aaa authentication attempts lockout max-fail 5 aaa authentication attempts lockout unlock-time 60 4. In a new SSH session, attempt to authenticate to the TOE via the CLI using an invalid password. 5. Verify that the authentication attempt failed. 6. Repeat Step 4 four additional times. 7. Attempt to authenticate to the TOE via the CLI using a valid password. 8. Verify that the authentication attempt failed due to account lockout. 9. Wait 60 seconds and then attempt to authenticate via the CLI using a valid password. 10. Verify that the authentication attempt succeeds. 11. Repeat Steps 3-11, except in Step 3 specify the max-fail value to 7 and the unlock-time value to 90, in Step 6 replace “four” with “six”, and in Step 11, replace “60” with “90”. 12. Verify that the authentication attempt succeeds.
Test Results	The evaluator confirmed the ability to configure the lockout maximum failure value and unlock-time value, the TOE successfully locking the offending remote user

	account that has met the lockout maximum failure value, the TOE unlocking the offending locked remote user account after the set unlock-time value is achieved, and the TOE successfully authenticating the unlocked remote user account when a valid password is entered - Pass
Execution Method	Manual

Test Case Number	063
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): 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 test assurance activity is tested in FIA_AFL.1 – Test Case 062.
Test Results	Pass
Execution Method	Manual

Test Case Number	064
SFR	FIA_PMG_EXT.1
Test Objective	The evaluator shall perform the following tests. 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 SSH.
	2. Enter the following commands to change the password of a user:
	
	`enable`
	`config terminal`
	`username cctl password`

	`abcdefghijklmnopqrstuvwxyzA12!` 3. In a new SSH session, authenticate to the TOE and attempt to login with the username and password that was configured in Step 2. 4. Verify that the authentication was successful. 5. Repeat Steps 1-4, except replace “`abcdefghijklmnopqrstuvwxyzA12!`” with “`BCDEFGHIJKLMNOPQRSTUVWXYZa345@`”. 6. Repeat Steps 1-4, except replace “`abcdefghijklmnopqrstuvwxyzA12!`” with “`aA67890#$%^&*()`”. 7. Repeat Steps 1-4, except replace “`abcdefghijklmnopqrstuvwxyzA12!`” with “`hijklA1!`”.
Test Results	The evaluator confirmed that attempts to change the password to values compliant with the password length requirement of at least 8 characters and containing all of the claimed characters were successful - Pass
Execution Method	Manual

Test Case Number	065
SFR	FIA_PMG_EXT.1
Test Objective	The evaluator shall perform the following tests. 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	**a) CLI:**
	1. Authenticate to the TOE via SSH.
	2. Enter the following commands to change the password of a user:
	`enable`
	`config terminal`
	`username cctl password bcdefgh`
	3. In a new SSH session, authenticate to the TOE and attempt to login with the username and password that was configured in Step 2.
	4. Verify that the authentication was unsuccessful.
	5. Repeat Steps 1-4, except replace “`bcdefgh`” with “`BCDEFG`”

Test Results	The evaluator confirmed that attempts to change the password to values less than 8 characters in length were unsuccessful - Pass
Execution Method	Manual

Test Case Number	066
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. 2. While entering password information, verify that the most obscured feedback is provided.
Test Results	The evaluator confirmed that the authentication feedback is obscured and not visible from the local console - Pass
Execution Method	Manual

Test Case Number	067
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: 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):** 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. **Remote SSH (password based):** 1. Authenticate to the TOE via SSH 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 SSH 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 SSH 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 SSH 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 (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.

LDAP Authentication (via the remote SSH CLI):

1. Authenticate to the TOE via SSH using a valid LDAP username and password:

   ```
   ssh testUser1@<TOE-IP-Address>
   ```

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 LDAP username and

	valid password. ssh testUserInvalid@<TOE-IP-Address> 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 LDAP username and an invalid password. ssh testUser1@<TOE-IP-Address> 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 LDAP username and an invalid password: ssh testUserInvalid@<TOE-IP-Address> 8. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure. **LDAP Authentication (via the local console CLI):** 1. Authenticate to the TOE via the local console using a valid LDAP username and password. testUser1 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 LDAP username and valid password. testUserInvalid 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 LDAP username and an invalid password. testUser1 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 LDAP username and an invalid password. testUserInvalid 8. Verify that the TOE failed to authenticate and that audit logs were generated reflecting the failure.
Test Results	The evaluator confirmed that for each set of valid credentials, the TOE successfully authenticates. For any set of credentials where any of the components are invalid, the TOE rejects the authentication attempt - Pass
Execution Method	Manual

Test Case Number	068
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: 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	Remote CLI 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 “show version” at the username and password prompts.
Test Results	The evaluator confirmed that the pre-authentication warning banner is the only service available prior to remote authentication - 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: 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	1. In a new console session, verify that the warning banner configured in the Setup is displayed prior to authentication to the TOE. 2. In a new console session, verify that no other services are available prior to authentication by entering a privileged command such as “show version” at the username and password prompts.
Test Results	The evaluator confirmed that the pre-authentication warning banner is the only service available prior to local authentication - 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: 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	N/A
Test Steps	N/A – The TOE is not a distributed TOE.
Test Results	N/A
Execution Method	N/A

Test Case Number	071
SFR	FIA_UIA_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	Per the assurance activity, evaluation activities for this requirement are covered under those for FIA_UIA_EXT.1.
Test Results	See FIA_UIA_EXT.1 Tests 67, 68, and 69 - Pass
Execution Method	Manual

Test Case Number	072
SFR	FIA_X509_EXT.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, and intermediate02 certificates on the remote server. 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 was established.

	6. Remove the root CA certificate from the TOE’s certificate authority trust store. 7. Repeat Steps 3-4 then jump to step 8. 8. Verify connection failed to establish.
Test Results	The evaluator confirmed that the TOE successfully completes the connection when all of the certificates are present in the trust store and the server sends the complete chain. Additionally, the evaluator confirmed that the TOE denies the connection when the intermediate 01 CA certificate was removed from the server presented certificate chain - Pass
Execution Method	Manual

Test Case Number	073
SFR	FIA_X509_EXT.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 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 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. 4. Verify connection failed to establish because of expired certificate.
Test Results	The evaluator confirmed that the TOE denied the connection to the remote server when the presented certificate's validity period was expired relative to the TOE's clock - Pass
Execution Method	Manual

Test Case Number	074
SFR	FIA_X509_EXT.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 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	TOE acting as a Client connecting to a Server CRL 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. Verify connection was established 6. Load a revoked server certificate onto the server. 7. Repeat Steps 2-4. 8. Verify connection failed to establish. 9. Load a valid server certificate onto the server. 10. Load a revoked intermediate01 CA certificate onto the server. 11. Repeat Steps 2-4. 12. Verify connection failed to establish.
Test Results	The evaluator confirmed that when none of the presented certificates are revoked, the TOE successfully establishes a connection to the remote server. Additionally, the evaluator confirmed the TOE rejects the connection when either the node certificate was revoked or when the intermediate 01 CA certificate was revoked - Pass
Execution Method	Manual

Test Case Number	075
SFR	FIA_X509_EXT.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 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**

	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). 3. Verify connection failed to establish.
Test Results	The evaluator confirmed 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 fails - Pass
Execution Method	Manual

Test Case Number	076
SFR	FIA_X509_EXT.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.1.1/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 TLS Client connecting to a Server 1. Begin capturing packets between the TOE and the environmental entity. 2. Run the modification test tool on the test system. 3. Cause the TOE to initiate a connection to the environmental entity. 4. Stop capturing packets between the TOE and the environmental entity. 5. Verify the connection failed to establish.
Test Results	The evaluator confirmed that the TOE fails to validate the certificate and denies the connection to the remote server when a single byte is modified in the first eight bytes of the presented certificate and the connection fails - Pass
Execution Method	Manual

Test Case Number	077
SFR	FIA_X509_EXT.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.1.1/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 TLS Client connecting to a Server 1. Begin capturing packets between the TOE and the environmental entity. 2. Run the modification test tool on the test system. 3. Cause the TOE to initiate a connection to the environmental entity. 4. Stop capturing packets between the TOE and the environmental entity. 5. Verify the connection failed to establish because the certificate signature will fail to validate.
Test Results	The evaluator confirmed that the TOE fails to validate the certificate when a single byte in the presented certificate signatureValue field is modified and the connection fails - Pass
Execution Method	Manual

Test Case Number	078
SFR	FIA_X509_EXT.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.1.1/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 TLS Client connecting to a Server 1. Begin capturing packets between the TOE and the environmental entity. 2. Run the modification test tool on the test system. 3. Cause the TOE to initiate a connection to the environmental entity. 4. Stop capturing packets between the TOE and the environmental entity. 5. Verify the connection failed to establish because the certificate hash will fail to validate.
Test Results	The evaluator confirmed that the TOE fails to validate the certificate when a single byte in the public key of the presented certificate is modified and the connection fails - Pass
Execution Method	Manual

Test Case Number	079
SFR	FIA_X509_EXT.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.1.1/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 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain. The evaluator shall replace the intermediate certificate in the certificate chain for Test 8 with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 8, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid. 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 TLS Client connecting to a Server 8a 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 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. Generate the explicit parameter version of the key generated from using a named curve: 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. 14. Verify that the TOE rejects the loading of the “int CA 01 explicit” certificate into the TOE’s trust store.

Test Results	The evaluator confirmed 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 confirmed that the TOE 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 confirmed that the TOE 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	080
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	1. For the LDAP TLS client interface, 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 failed to establish.
Test Results	The evaluator confirmed that the TOE 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 contain the basicConstraints extension and the connection fails - Pass
Execution Method	Manual

Test Case Number	081
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 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	1. For the LDAP TLS client interface, 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 failed to establish.
Test Results	The evaluator confirmed that the TOE rejects the certificate when the intermediate 02 CA in the presented chain does not have the CA flag value set to TRUE and the connection fails - Pass
Execution Method	Manual

Test Case Number	082
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 1. Begin capturing packets between the TOE and the environmental entity 2. Initiate a connection from the TOE to the server. 3. Verify the connection succeeds. 4. Remove the intermediate02 CRL from the distribution point. 5. Begin capturing packets between the TOE and the environmental entity 6. Initiate a connection from the TOE to the server. 7. Verify the connection failed to establish due to the TOE being unable to verify the certificate.
Test Results	The evaluator confirmed that when the TOE is able to successfully communicate with the CRL distribution point and receives a valid CRL, the TOE successfully establishes a connection to the remote server. Additionally, the evaluator confirmed that when the TOE is unable to successfully communicate with the CRL distribution point and receive a valid CRL, the TOE denies the connection to the remote server, which is consistent with the ST selection for this SFR - Pass
Execution Method	Manual

Test Case Number	083
SFR	FIA_X509_EXT.3
Test Objective	The evaluator shall perform the following tests: Test 1: The evaluator shall use the guidance documentation to cause the TOE to generate a Certificate Request. The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that

	the Certificate 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	1. Authenticate to the TOE via the CLI. 2. Execute the following commands to generate a certificate request message: enable config terminal crypto cert-req-msg generate upload scp://username@SCP_Server:/path/filename 3. Enter the remote SCP server password. 4. On the remote SCP server, execute the following command to verify the Certificate Request contains the public key, Common Name, Organization, Organizational Unit, and Country: openssl req -in <uploaded-csr-filename>.csr -noout -text
Test Results	The TOE successfully created a certificate request message with the required information The evaluator was able to successfully validate the created CSR - Pass
Execution Method	Manual

Test Case Number	084
SFR	FIA_X509_EXT.3
Test Objective	The evaluator shall perform the following tests: 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. Authenticate to the TOE via the CLI. 2. Execute the following commands to generate a certificate request message: enable config terminal crypto cert-req-msg generate upload scp://username@SCP_Server:/path/filename 3. Sign the certificate request message. 4. Transfer the signed certificate without a valid certificate path to the TOE. 5. Execute the following command to access the TOE debug shell: debug shell req 6. Provide the challenge phrase to the vendor. 7. Execute the following command to provide the challenge response from

	the vendor to access the debug shell: debug shell enter <response code> 8. Execute the following command on the TOE to validate the signed certificate: openssl verify -CAfile invalid_chain.pem 080.cert 9. Transfer the signed certificate with a valid certificate path to the TOE. 10. Execute the following command on the TOE to validate the signed certificate: openssl verify -CAfile valid_chain.pem 080.cert
Test Results	The evaluator observed that a CSR without CAs installed (i.e. invalid certification path) failed validation. The evaluator observed that a CSR with the proper CAs installed validated correctly - Pass
Execution Method	Manual

4.4.4 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 CLI as ‘limiteduser’ user. 2. Follow the update procedures described in FPT_TUD_EXT.1 – Test Case 092 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 092.
Test Results	The evaluator confirmed that a limited user account, "limiteduser" (non-security administrator) does not have sufficient permissions to update the TOE software as the command used to update the TOE was not recognized as a valid command while logged in as a limited user. Additionally, see FPT_TUD_EXT.1.1 for the successful attempt to initiate an update - 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 ‘limiteduser’. 2. Execute the following commands to generate new SSH server host keys on the TOE: enable config terminal ssh server host-key generate 3. Verify that the “ssh server host-key generate” command failed to execute. 4. Log out of the TOE. 5. Authenticate to the TOE via the CLI as the Security Administrator (i.e. admin). 6. Repeat Step 2. 7. Verify that the “ssh server host-key generate” command executed successfully.
Test Results	The evaluator confirmed that a limited user account, "limiteduser" (non-security administrator) does not have sufficient permissions to manage the TOE crypto configuration - 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 SFR assurance activity is satisfied by the testing of other SFRs in this test plan.
Test Results	The evaluator confirmed that all functions claimed in the FMT_SMF.1 have been tested in the course of performing other test cases - Pass
Execution Method	Manual

Test Case Number	088
SFR	FMT_SMR.2
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 SFR assurance activity is satisfied by the testing of other SFRs in this test plan.
Test Results	The evaluator confirmed that all administrative user interfaces were used in the course of testing and functions claimed in the FMT_SMF.1 have been tested in the course of performing other test cases - Pass
Execution Method	Manual

4.4.5 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	CLI 1. Authenticate to the TOE via SSH. 2. Enter the following commands to set the date and time: enable config terminal clock set <hh:mm:ss> [<yyyy/mm/dd>] 3. Enter the following command to verify that the time and date were set to the values specified in Step 2: show clock
Test Results	The evaluator confirmed the ability to manually configure the TOE's clock and that the TOE implemented the requested change successfully - 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	N/A
Test Steps	N/A – Per the Security Target, NTP is not claimed; therefore, this test does not apply.
Test Results	N/A
Execution Method	N/A

Test Case Number	109
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	N/A
Test Steps	Per the Security Target, time is not obtained from VS
Test Results	N/A
Execution Method	N/A

Test Case Number	091
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 local console. 2. Enter the following commands to reboot the TOE: enable config terminal reload 3. Verify that the TOE performs an integrity check of the firmware and executable software of the TOE. 4. Verify that the TOE verifies the correct operation of its cryptographic functionality.
Test Results	The evaluator confirmed that the TOE correctly performed integrity checks of its firmware and executables software at boot time. The TOE also performed self-tests of its cryptographic functions as expected - Pass
Execution Method	Manual

Test Case Number	092
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 CLI. 2. Execute the following commands to obtain the current and most recently installed TOE version:

	`enable`
	`config terminal`
	`show version`
	
	3. Execute the following commands to fetch and initiate the TOE software update:
	
	`enable`
	`config terminal`
	` image fetch [PROTOCOL]://[IP-ADDRESS]/[FILE]`
	
	`image install <image file>`
	
	4. Prior to activation of update, confirm the TOE version corresponds to the current version:
	
	`show version`
	5. Activate the most recently installed update by executing the following commands:
	
	`image boot next`
	`reload`
	
	6. After the TOE fully boots, 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 confirmed that the TOE's version, prior to the successful update attempt using a valid update, properly increased after the TOE correctly applied the valid update - Pass
Execution Method	Manual

Test Case Number	093
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	Modify a legitimate update with a hex editor such that the integrity of the update is compromised, producing an illegitimate update. Modify the update such that it is not signed: 1. Extract the contents of the supplied update file. 2. Inspect the contents and identify the core update file / package. 3. Calculate the SHA256 hash of the core update file / package and confirm it matches the SHA256 hash recorded in the manifest. 4. Remove the PGP signature data from the manifest's signature file 5. Repackage the contents into the supplied update file. Modify the signature of the update: 6. Extract the contents of the supplied update file. 7. Inspect the contents and identify the core update file / package. 8. Calculate the SHA256 hash of the core update file / package and confirm it matches the SHA256 hash recorded in the manifest. 9. Modify the PGP signature from the manifest's signature file: a. Convert the base64 encoded data to hexadecimal b. Modify a random section of the hexadecimal values such that it is sufficiently modified c. Re-encode the modified data to base64 10. Repackage the contents into the supplied update file.
Test Results	The evaluator confirmed that the TOE version, prior to failed update attempts using invalid updates (modified binary via hex edit, missing signature, modified signature) are presented to the TOE, remains the same and that the TOE did not install the invalid updates - Pass
Execution Method	Manual

Test Case Number	094
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	N/A

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	N/A
Execution Method	N/A

4.4.6 TOE Access

Test Case Number	095
SFR	FTA_SSL_EXT.1
Test Objective	The evaluator shall perform the following test: 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. 2. Enter the following commands to configure the inactivity time period for session termination: enable config terminal cli default auto-logout 3 3. Exit the session and then in a new session, authenticate to the TOE via the local console. 4. Do not perform any action for 3 minutes. 5. Immediately after 3 minutes have elapsed, verify that the local session has been terminated. 6. Repeat Steps 1-5, except replace “3” with “5.” 7. Repeat Steps 1-5, except replace “3” with “7.”
Test Results	The evaluator confirmed the ability to configure the inactivity timeout value, the TOE successfully terminates the local session at the set interval, and that audit records are produced for the inactivity termination of the session - Pass
Execution Method	Manual

Test Case Number	096
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 SSH. 2. Enter the following commands to configure the inactivity time period for session termination: enable config terminal cli default auto-logout 4 3. Exit the session and then in a new session, authenticate to the TOE via SSH. 4. Do not perform any action for 4 minutes. 5. Immediately after 4 minutes have elapsed, verify that the SSH session has been terminated. 6. Repeat Steps 1-5, except replace "4" with "6". 7. Repeat Steps 1-5, except replace "4" with "8".
Test Results	The evaluator confirmed the ability to configure the inactivity timeout value, the TOE successfully terminates the remote session at the set interval, and that audit records are produced for the inactivity termination of the session - Pass
Execution Method	Manual

Test Case Number	097
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. 2. Enter the "exit" command to terminate the session. 3. Observe that the session has been terminated. 4. Confirm that the session was terminated by attempting to enter privileged commands such as "show version".
Test Results	The evaluator confirmed the ability to terminate one's own local session and that audit records are produced for the user-initiated termination of the CLI session - Pass
Execution Method	Manual

Test Case Number	098
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. Enter the "exit" command to terminate the session.

	3. Observe that the session has been terminated.
Test Results	The evaluator confirmed the ability to terminate one's own remote session and that audit records are produced for the user-initiated termination of the CLI session - Pass
Execution Method	Manual

Test Case Number	099
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	Remote CLI 1. Authenticate to the TOE via SSH. 2. Enter the following commands to configure the warning banner: enable config terminal banner login “!!THIS IS A WARNING BANNER!!” 3. In a new SSH session, verify that the warning banner configured in Step 2 is displayed prior to authentication to the TOE. Local CLI 4. Authenticate to the TOE via the local console. 5. Enter the following commands to configure the warning banner: enable config terminal banner login “!!THIS IS A NEW WARNING BANNER!!” 6. In a new local console session, verify that the warning banner configured in Step 5 is displayed prior to authentication to the TOE.
Test Results	The evaluator confirmed the ability to configure a warning banner and that the warning banner was displayed on all of the claimed interfaces used for authentication to the TOE (local console, remote SSH) - Pass
Execution Method	Manual

4.4.7 Trusted Path/Channels

Test Case Number	100
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. 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	**a) TOE and Remote LDAP Server** 1. Begin capturing packets between the TOE and the LDAP server. 2. Initiate a connection to the LDAP server from the TOE by using LDAP credentials on the TOE. 3. Stop capturing packets between the TOE and the LDAP server. 4. Examine the packet capture and verify the data transmitted between the TOE and LDAP server are protected using TLS. **b) TOE and Remote syslog Server** 1. Begin capturing packets between the TOE and the syslog server. 2. On the TOE, perform an action that causes the TOE to initiate a connection to the syslog server by performing an action that causes an audit record to be transmitted to the syslog server. 3. Stop capturing packets between the TOE and the syslog server. 4. Examine the packet capture and verify the data transmitted between the TOE and syslog server are protected using SSH. **a) FM and GigaVUE Machine** 1. Begin capturing packets between the FM device and the GigaVUE machine. 2. On the FM device, perform an action that causes the TOE to initiate a connection to the GigaVUE machine by performing an action that causes an audit record to be transmitted to the syslog server. 3. Stop capturing packets between the FM device and the GigaVUE machine. 4. Examine the packet capture and verify the data transmitted between the FM device and the GigaVUE machine are protected using SSH.
Test Results	The evaluator confirmed that a trusted channel, via TLS, was successfully initiated

	by the TOE to the LDAP server and channel data was not sent in plaintext. The evaluator confirmed that a trusted channel, via SSH, was successfully initiated by the TOE to the syslog server and channel data was not sent in plaintext. The evaluator confirmed that the trusted channel to the TOE from the Fabric Manager was successfully established and the channel data was not sent in plaintext - Pass
Execution Method	Manual

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 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. 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	Testing of this assurance activity is performed using FTP_ITC.1 – Test Case 100.
Test Results	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 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. 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	Testing of this assurance activity is performed in FTP_ITC.1 – Test Case 100.
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 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	**TOE and Remote LDAP Server** 1. Begin capturing packets between the TOE and the LDAP server. 2. Physically disconnect the connection between the TOE and the LDAP server. 3. Initiate the connection to the LDAP server from the TOE by authenticating to the TOE using LDAP credentials via the local console: testUser1 4. Restore the connection between the TOE and the LDAP server no sooner than 10 seconds so the connection times out. 5. Repeat step 3. 6. Stop capturing packets between the TOE and the LDAP server. 7. Examine the packet capture and verify the data transmitted between the TOE and LDAP server are protected using TLS. 8. Repeat Steps 1-4, except in Step 4, replace 10 seconds with 2 seconds. 9. Stop capturing packets between the TOE and the LDAP server. 10. Examine the packet capture and verify the data transmitted between the TOE and LDAP server are protected using TLS. **TOE and Remote syslog Server** 1. Begin capturing packets between the TOE and the syslog server. 2. On the TOE, perform an action that causes the TOE to initiate a connection to the syslog server by performing an action that causes an audit record to be transmitted to the syslog server. (Execute on TOE): enable config terminal config write 3. Physically disconnect the connection between the TOE and the syslog server. 4. Restore the connection between the TOE and the syslog server no sooner than 60 seconds. 5. Repeat Step 2. 6. Stop capturing packets between the TOE and the syslog server. 7. Examine the packet capture and verify the data transmitted between the TOE and syslog server are protected using SSH. 8. Repeat Steps 1-4, except in Step 4, replace 60 seconds with 15 seconds. 9. Stop capturing packets between the TOE and the syslog server. 10. Examine the packet capture and verify the data transmitted between the TOE and syslog server are protected using SSH. **FM TO TOE (not applicable to GTAP)**

	1. Begin capturing packets between the TOE and the FM Machine 2. Initiate the connection to the LDAP server from the TOE by authenticating to the TOE using LDAP credentials via the local console: Username: admin Password: P@ssword1234!@#$ 3. Physically disconnect the connection between the TOE and the FM machine. 4. Restore the connection after no sooner than 10 seconds. 5. Verify connection fails for the 10 second disconnect. 6. Stop capturing packets between the TOE and the FM machine and verify the first attempt fails because of no connectivity and verify the second attempt succeeds. Perform a search for the password in the packet capture and ensure the password does not show up. 7. Begin capturing packets between the TOE and the FM machine. 8. Repeat steps 2-5 but instead of a 10 second disconnect change the time to a 2 second disconnect. 9. Verify the connection succeeds. 10. Stop capturing packets between the TOE and the FM machine. 11. Examine the packet capture and verify the data transmitted between the TOE and FM machine are protected using TLS and perform a search for the password in the packet capture and ensure the password does not show up.
Test Results	The evaluator confirmed that the TOE successfully re-established a secure channel to the LDAP server (TLS), syslog server (SSH), and Fabric Manager Server after the connection was re-established and no data was passed in plaintext - Pass
Execution Method	Manual

Test Case Number	104
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	**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.

Test Results	The evaluator confirmed that a trusted path, via SSH, was successfully established, able to be used for remote administration using the TOE's CLI, and channel data was not 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 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	This test assurance activity is met by testing performed in FTP_TRP.1 – Test Case 104.
Test Results	Pass
Execution Method	Manual

4.4.8 Conclusion of IND testing

In all cases of the independent testing, the expected functionality was observed, and the actual results were found to be consistent with the expected results. The evaluators determined that the SFRs and ST claims were thoroughly tested, and the product performed as expected. Comparison of the test results of all models verified that differences in CPU, processors, binary images, and subset of functional claims did not result in differences between the same functionality claimed between these models. Additionally, the full testing of the HC1, TA200, and GTAP with the additional sampling performed on the HC3, HC1Plus, HCT, TA25, TA25E, TA200E, and TA400 further confirmed that the claim of ‘equivalent functionality even though it uses a different installation binary’ is accurate. The IND assurance activity is considered satisfied as the required testing has been performed successfully.

5 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.

NOTE: Any distributed TOE assurance activities were omitted below since the TOE is not a distributed TOE.

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."*

Section 1.3 of the Security Target describes the purpose and method of use for each security relevant TSFI by enumerating all security relevant interfaces:

> E1: This is the local administrator access to the CLI via a direct connection.
> E2: The TOE acts as a SSH server for remote administrator access to the CLI.
> E3: The TOE acts as a TLSv1.2 client for accessing an LDAP server interface for authentication services.
> E4: The TOE acts as an SSH client for sending audit records to a remote audit server for external audit log storage.
> E5: (HC Series and TA Series models only) The TOE acts as a HTTPS (i.e., TLSv1.2) server for connections received from a Gigamon Fabric Manager (separate product) which can be used to provide a central location for the configuration, management, and operation of primary functionality of one or more Gigamon GigaVUE HC and TA Visibility Appliances. The trusted channel interface is considered part of the TOE. The operational functionality provided by the Gigamon Fabric Manager is not considered part of the TOE.
> E6: The TOE interfaces with a Certification Authority (CA) for issuance of server certificates and publication of a Certificate Revocation List (CRL) to determine the validity of certificates presented to the TOE.

The list also clearly identifies those interfaces out of NDcPP testing:

> E7-E13: These data plane interfaces are used for GigaVUE's primary functionality of forwarding and copying network data to one or many tool ports for packet capture or analyzing tools. The operational functionality performed over the data plane do not map to any NDcPP2E security requirements. Therefore, the non-interfering interfaces and the functionality they provide are not in-scope of the evaluation.

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."*

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. 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."*

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: These interfaces are used to perform management actions on the TOE. Each management action will generate an audit log with the identity of user. (GEN.1 and GEN.2)
E4: This interface is used for external audit storage via a Syslog server.

Cryptographic Support (FCS)
E2: Remote administration authentication (password and public key) and TSF Data is sent over this interface and is protected with SSHv2. (SSHS_EXT.1)
E3: Authentication Data sent to the LDAP server over this interface is protected with TLSv1.2. (TLSC_EXT.1, FCS_CKM.2, FCS_COP. as applicable to ciphers)
E4: Audit data sent over this interface is protected by SSHv2 and Public Key authentication to the syslog server (SSHC_EXT.1, FCS_ COP.1/XX as applicable to ciphers)
E5: Connection with Gigamon Fabric Manager (HTTPS_EXT.1, FCS_TLSS_EXT.1)
E6: Certificate revocation checking is performed over this interface. Certificates are used for TLS connections to LDAP. (TLSC_EXT.1)

Identification and Authentication (FIA)
E1, E2: 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, UAU.7, AFL.1, PMG_EXT.1)

E3: Authentication information for LDAP authentication is transmitted over this interface. (UIA_EXT.1, UAU_EXT.2)
E6: Certificate revocation checking is performed over this interface. Certificates are used for TLS connections to LDAP. (X509_EXT.1 and X509_EXT.2)

Security Management (FMT)
E1, E2: 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: All management actions are performed over these interfaces. (FPT_STM_EXT.1)

TOE Access (FTA)
E1, E2: All 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, SSL_EXT.1, TAB.1)

Trusted Path/Channels (FTP)
E2: Remote Administration data sent over this interface is protected with SSHv2 (FTP_TRP.1/Admin)
E3: LDAP data sent over this interface is protected with TLS 1.2 (FTP_ITC.1)
E4: Audit data sent over this interface is protected with SSHv2 (FTP_ITC.1)
E5: Data sent over this interface is protected with TLS 1.2 (FTP_ITC.1)

AGD_OPE.1 – *"The evaluator shall ensure the Operational guidance documentation is distributed to Security Administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that Security Administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration."*

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 *Gigamon GigaVUE Version 6.5 Supplemental Administrative Guidance (AGD)* document contains configuration instructions for placing the TOE in its evaluated configuration. Additionally, as part of the CC certification process, the AGD is published on the NIAP web site supplementing the vendor guidance documentation. Therefore, there is a reasonable guarantee that administrators and users are aware of this documentation due to its listing on the Product Complaint List (PCL) in conjunction with the certified product.

"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."

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 Gigamon GigaVUE version-OS 6.5. 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 Gigamon GigaVUE-OS Version 6.5 and the general CC terminology that is referenced in it. This supplemental guidance includes references to Gigamon GigaVUE’s standard documentation set for the product and does not explicitly reproduce materials located there.” Tables 1, 2, and 3 in the AGD and Tables 2-3, 2-4, and 2-5 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.”

Section 6.3 of the AGD states “The administrator installing the TOE is expected to perform all of the operations in Sections 6.1 through 6.5 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 the TOE already becomes pre-configured to meet many of the Common Criteria requirements such as limiting all ciphersuites and algorithms to those defined in the Security Target [1] and automatic zeroization key destruction functionality.” The description goes on to warn the reader: “NOTE: The use of other cryptographic engines and cryptographic settings were not evaluated nor tested during the Common Criteria evaluation of the TOE.”

“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.”

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.
Thus, the evaluation team has determined that only the commands and interfaces described within the AGD, as well as the specific pointers in the AGD to other documents, are considered to be security relevant for this evaluation. Section 7 of the AGD indicates that the “The following sections provide information on managing TOE functionality that is relevant to the claimed Protection Profile.” The evaluator found there

was a one-to-one correspondence with the sections in the AGD and the defined Security Administrator functionality defined in the ST.

"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.*
>
> Section 6.3 provides instructions for the administrator to configure the TOE to use the Secure Cryptography Mode. The description also states, "There is no further configuration required on the TOE's cryptographic engine as the TOE already becomes pre-configured to meet many of the Common Criteria requirements such as limiting all ciphersuites and algorithms to those defined in the Security Target and automatic zeroization key destruction functionality." The description goes on to warn the reader: "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]
>
> > *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).*
> >
> > *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.*
>
> 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*

[5] TD0536

> *an administrator which security functionality is covered by the Evaluation Activities."*

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 GigaVUE 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 Gigamon GigaVUE 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)."*

Section 5.3 of the AGD contains instructions for the Security Administrator to ensure that the operational environment will fulfil its role in supporting the TOE. These instructions match the assumptions for the TOE's operational environment in Section 4.3 of the ST.

"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."

The evaluators determined from a review of the ST that the TOE has 14 models. All models operate using the identical Gigamon GigaVUE-OS Version 6.5. The evaluators observed from conducting the Evaluation Activities for the operational guidance that the supplemental AGD includes and/or references sufficient information to describe how to manage the TSF. The evaluators also observed that the supplemental AGD references the installation guidance that is relevant to all TOE models. The installation documentation suite also includes individual specific hardware installation manuals for the different models.

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

Tables 1, 2 and 3 in the AGD and Tables 2-3, 2-4 and 2-5 in the ST match and describe the only TOE model 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 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."

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 Gigamon GigaVUE-OS Version 6.5. 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 Gigamon GigaVUE-OS Version 6.5 and the general CC terminology that is referenced in it. This supplemental guidance includes references to Gigamon GigaVUE's standard documentation set for the product and does not explicitly reproduce materials located there. 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."

Tables 1, 2, and 3 in the AGD and Tables 2-3, 2-4, and 2-5 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. Since all of the models use the same GigaVUE-OS software, the instructions provided in the AGD apply to all models and encompass all of the necessary steps to securely manage and of the TOEs in the installed environment. The AGD's procedures were used to successfully perform the required testing of the TOE in its evaluated configuration. 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.

"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."

When the TOE has been installed and configured as specified in the administrative guidance the TOE provides the protected administrative capabilities. The documentation clearly describes the role-based management capabilities that is enforced on the TOE. The assumptions of use also contain the expectation that the administrators will protect their passwords for unauthorized disclosures. The AGD contains all of the instructions necessary to configure the TOE to support public key authentication for SSH connections. Secure channels use of SSH are automatically supported and cannot be turned off.

Section 6.1 of the AGD describes initial TOE installation default credentials with a warning to modify the default password for the 'admin' account. During the installation, the TOE forces the user to change the default password to a non-default password. The default password (admin123A!) will never be accepted as a valid password in any future attempts to change the password.

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 "the Gigamon GigaVUE Version 6.5 family of products, which includes the following appliance models: GigaVUE-HC Series, and GigaVUE-TA Series, and GigaTAP A Series. Each appliance runs the Gigamon GigaVUE-OS software Version 6.5. The TOE software version was queried by executing the "show version" command from the CLI. The TOE hardware was identified by physical examination of the network appliance.

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."*

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 "the Gigamon GigaVUE Version 6.5 family of products, which includes the following appliance models:" GigaVUE-HC Series, GigaVUE-TA Series, and GigaTAP A Series. Each appliance runs the Gigamon GigaVUE-OS software version 6.5. The TOE software version was queried by executing the "show version" command from the CLI. The TOE hardware was identified by physical examination of the network appliance.

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."

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

Keyword	Description
Gigamon	This is a generic term for searching for known vulnerabilities produced by the company as a whole.
GigaVUE (6.5)	This is a generic term for searching for known vulnerabilities for the specific product which will cover GigaVUE-OS
Rocky Linux (8.7)	This is a generic term searching for known vulnerabilities for the underlying operating system.
Libraries	
OpenSSL (3.0.14B)	This is a generic term searching for known vulnerabilities for the TOE’s cryptographic TLS module.
OpenSSH (9.8p1)	This is a generic term searching for known vulnerabilities for the TOE’s cryptographic SSH module.
curl (8.9.1)	This is a generic term searching for known vulnerabilities for the third-party library.
bind-export-libs (9.11.36-14)	This is a generic term searching for known vulnerabilities for the third-party library.
bind-libs-lite (9.11.36-14)	This is a generic term searching for known vulnerabilities for the third-party library.
bzip2 (1.0.6-26)	This is a generic term searching for known vulnerabilities for the third-party library.
grub2 (2.02-156)	This is a generic term searching for known vulnerabilities for the third-party library.
gzip (1.9-13)	This is a generic term searching for known vulnerabilities for the third-party library.
log4cxx (0.10.0-31)	This is a generic term searching for known vulnerabilities for the third-party library.
ldap (2.6.4)	This is a generic term searching for known vulnerabilities for the third-party library.
Perl (5.26.3-421)	This is a generic term searching for known vulnerabilities for the third-party library.
rsyslog (8.2102.0-10)	This is a generic term searching for known vulnerabilities for the third-party library.
SQLite (3.26.0-19)	This is a generic term searching for known vulnerabilities for the third-party library.
unzip (6.0-46)	This is a generic term searching for known vulnerabilities for the third-party library.
zlib (1.2.11-20)	This is a generic term searching for known vulnerabilities for the third-party library.
Hardware	

Keyword	Description
Intel Atom (C2758 and C2538) (Rangely)	This is a generic term searching for known vulnerabilities for the TOE's underlying host processor.
Intel Xeon (D1527) (Broadwell)	This is a generic term searching for known vulnerabilities for the TOE's underlying host processor.
4.4.3.3 Keyword: Intel Xeon (D1518)	This is a generic term searching for known vulnerabilities for the TOE's underlying host processor.
Intel Atom (C3338 and 3538) (Denverton)	This is a generic term searching for known vulnerabilities for the TOE's underlying host processor.

These keywords were used individually and as part of various permutations and combinations to search for vulnerabilities on public vulnerability sources (updated November 15, 2024). The following public vulnerability sources were searched:

- NIST National Vulnerabilities: https://web.nvd.nist.gov/view/vuln/search
- Common Vulnerabilities and Exposures: http://cve.mitre.org/cve/ https://www.cvedetails.com/vulnerability-search.php
- US-CERT: http://www.kb.cert.org/vuls/html/search
- Tenable Network Security http://nessus.org/plugins/index.php?view=search
- Tipping Point Zero Day Initiative http://www.zerodayinitiative.com/advisories
- Offensive Security Exploit Database: https://www.exploit-db.com/
- Rapid7 Vulnerability Database: https://www.rapid7.com/db/vulnerabilities

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 enumerates network port and service information to determine if any ports were open and running services outside of the TOE standard configuration.
- Fuzzing – Mutated TYPE and CODE
  This test determines if the TOE is adversely affected by the handling of large number of mutated IPv4 and ICMPv4. IPv6 was not supported in the evaluated configuration.
- Fuzzing – Mutated remaining field
  This test determines if the TOE is adversely affected by the handling of large number of mutated IPv4 packets where the header fields are carefully mutated to represent boundary cases, significant values, and randomly chosen values. IPv6 was not supported in the evaluated configuration.
- SSH Timing Attack (User Enumeration)

This attack attempts to enumerate validate usernames for the SSH interface, by observing the difference in server response times to valid username login attempts.

- Force SSHv1
  This attack determines if the client will accept both SSHv1 and SSHv2 connections when the TOE claims to only support SSHv2
- CLI Privilege Escalation
  This attack involves enumerating a valid username with an attempt to access the underlying OS CLI shell, then cracking the user’s password and logging in.

The evaluation team determined that no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.

6 Conclusions

The evaluation was carried out in accordance with the Common Criteria Evaluation and Validation Scheme (CCEVS) processes and procedures. Gigamon GigaVUE Version 6.5 was evaluated against the criteria contained in the Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 5. The evaluation methodology used by the evaluation team to conduct the evaluation is the Common Methodology for Information Technology Security Evaluation, Version 3.1 Revision 5. The product, when installed and configured per the instructions provided in the preparative guidance, satisfies all of the security functional requirements stated in the *Gigamon GigaVUE Version 6.5 Security Target Version 1.0* as scoped by the NDcPP2.2E.

The overall verdict for this evaluation is: Pass.

7 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
IDS	Intrusion Detection System
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
NIAP	National Information Assurance Partnership
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OS	Operating System
PP	Protection Profile
RAM	Random Access Memory

RBG	Random Bit Generator
RNG	Random Number Generator
RU	Rack Unit
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SPAN	Switched Port Analyzer
SSH	Secure Shell
ST	Security Target
TAP	Test Access Port
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
UART	Universal Asynchronous Receiver/Transmitter
UI	User Interface

Table 7-1: 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 of 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 7-2: Terminology