Assurance Activity Report for
MMA10G-EXE Series II

MMA10G-EXE Series II Security Target
Version 1.2

collaborative Protection Profile for Network Devices, Version 2.2e

AAR Version 1.1, August 13,2024

Evaluated by:

2400 Research Blvd, Suite 395
Rockville, MD 20850

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:
Evertz Microsystems Ltd.

The Author of the Security Target:
Acumen Security LLC

The TOE Evaluation was Sponsored by:
Evertz Microsystems Ltd.

Evaluation Personnel:
Shehan D Dissanayake
Ashish Panchal

Common Criteria Version
Common Criteria Version 3.1 Revision 5

Common Evaluation Methodology Version
CEM Version 3.1 Revision 5

REVISION HISTORY

VERSION	DATE	CHANGES
1.0	17/07/2024	Initial Release
1.1	13/08/2024	Release after addressing NIAP validator comments. Updates to table 1, section 6.2.1.2, and section 8.1 to remove 3 models from the claimed models list

CONTENTS

1 TOE OVERVIEW

The MMA10G-EXE Series II switches are Internet Protocol (IP) switches optimized for video-over-IP traffic (compressed or uncompressed). The TOE is classified as a network device (a generic infrastructure device that can be connected to a network). Models of the EXE included in the evaluation provide identical functionality. The only differences between them are the supported speed, the physical size, and the number of physical interfaces supported, and the processor. These differences are detailed at the end of this section.

The EXE builds on the capabilities of the existing Evertz line of video routing switches. Video routers receive video signals in various formats, such as Serial Digital Interface (SDI), Serial Data Transport Interface (SDTI), or Asynchronous Serial Interface (ASI), and switch dedicated physical input ports to dedicated physical output ports based on external commands. The EXE provides the same capability within the context of packet-based networks using shared network infrastructure.

The TOE provides a packet-based switching fabric from a video perspective, rather than relying on traditional packet-based network architecture.

A typical EXE installation will also include a standard video routing switch software platform (such as Evertz Magnum) to route data between program streams in a manner sufficient to meet broadcast video standards for signal availability and integrity. Equipment to prepare video for IP transport, or to convert it into other video formats, and non-network-based video switching/processing, is outside the scope of this TOE. Such equipment includes, but is not limited to, cameras, KVMs, codecs, video servers and video displays. Equipment to perform functions such as embedding audio and/or other information within the video stream is also outside the scope of this TOE.

The TOE provides secure remote management using an HTTPS/TLS web interface. Administrators only may access EXE via a dedicated management workstation operating over an Out-of-Band Management (OOBM) network. Sites may close this OOBM network or may operate EXE within an existing OOBM as long as the topology is compliant with the security parameters listed below. Users and administrators may also access EXE software via direct connection using a terminal session.

The TOE generates audit logs and transmits the audit logs to a remote syslog server over an authenticated TLS channel. The TOE verifies the authenticity of software updates by verifying the digital signature prior to installing any update.

The summary of the evaluated functionality provided by the TOE includes the following,

- Secure connectivity with remote audit servers and secure retention of audit logs locally
- Identification and authentication of the administrator of the TOE
- Secure remote administration of the TOE via TLS and secure Local administration of the TOE
- Secure access to the management functionality of the TOE

- Secure software updates
- Secure communication with the non-TOE 'video switch control systems' via TLS.

The TOE hardware devices are the Evertz:

Table 1 – TOE hardware models

Model	AV/ Broadcast	Supported Ports	Form Factor	Chassis Supported	Frame Controller	Processor
NATX-8-100G-CC	broadcast	4 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core (TM) i3-6102E
NATX-16-100G-CC	broadcast	8 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core (TM) i3-6102E
NATX-32-100G-1-CC	broadcast	16 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core (TM) i3-6102E
NATX-64-100G-2-CC	broadcast	32 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core (TM) i3-6102E
MMA10G-NATX-8-CC	AV	4 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core (TM) i3-6102E
MMA10G-NATX-16-CC	AV	8 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core (TM) i3-6102E
MMA10G-NATX-32-CC	AV	16 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core (TM) i3-6102E
MMA10G-NATX-64-CC	AV	32 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core (TM) i3-6102E
MMA10G-IPX128	AV	32 x QSFP+	3 or 6	EV Frame	ev3-FC or ev6-FC	Intel(R) Core (TM) i3-6102E
3080IPX-48-25G-CC	AV/ broadcast	12 x QSFP+	3 or 6	EV Frame	ev3-FC or ev6-FC	Intel(R) Core (TM) i3-6102E

The EXE firmware version 1.5 will be referred to as EXE throughout this document.

The EXE appliances are Ethernet switches optimized for video content.

2 ASSURANCE ACTIVITIES IDENTIFICATION

The Assurance Activities contained within this document include all those defined within the NDcPP 2.2e based upon the core SFRs and those implemented based on selections within the PP.

3 TEST EQUIVALENCY JUSTIFICATION

3.1 HARDWARE.

The TOE chassis include:

- EV Frame
- DragonFire Frame

The EXE frames include 3 different form factors (16, 26, and 36).

MMA10G-IPX128 and 3080IPX-48-25G-CC models run on EV frames and have ev3 or ev6 Frame Controllers. They are on frames (chassis) that support form factor 3 (ev3-FC) and form factor 6 (ev6-FC).

The MMA10G-NATX models and NATX model chassis (Dragonfire frames) includes frame management within the chassis and provides the EXE card with access to ethernet interfaces. All models running on DragonFire frames have form factor 1.

Both EV frames and DragonFire frames come with a controller card that manages chassis function and provides the EXE card with access to ethernet interfaces. The frame controller includes one dummy L2 switch chip which is not accessible externally and is used to forward management traffic to MMA10G or 3080IPX device.

Although the chassis differs, the differences do not affect the functionality of the TOE. MMA10G firmware does not do any frame management. The 'Intel(R) Core (TM) i3-6102E C' is the only processors used across the claimed platforms.

3.2 DIFFERENCE IN TOE SOFTWARE BINARIES

Each product is uniquely identified by using unique disk images per product, which comprises of following common elements:

1. Bootloader.
2. Bootloader config, which includes unique product types.
3. An image that includes support for the product type. (NOTE: This is the same image used for testing. One image includes support for all the product types listed on the supported model).

TOE startup sequence is the Bootloader(1) would run the image(3) with the unique product type from the bootloader config(2). During the bootup time, the image(3) code probes the hardware(i.e., registers, etc.) it is booting in and will precisely identify the product type.

Once the product type is precisely identified during the bootup time, following are performed:

1. Set environment variables that identifies the product type precisely.
2. When applications inside the image(3) start, they use the environment variable and precisely enable product type specific features, i.e., number of ports, etc.

Due to above mentioned common logic, the source code variations are related only to hardware differences between the EXE cards. The source code for the TSF is identical across all models.

3.3 DIFFERENCES IN LIBRARIES USED TO PROVIDE TOE FUNCTIONALITY

There are no differences in the TOE libraries that provide the TOE functionality. The TOE leverages third party software which is the same for all models.

3.4 TOE MANAGEMENT INTERFACE DIFFERENCES

TOE management interface is provided by secure TLS v1.2 session or via a local console connection. The management interface is the same across all platforms, and the protocol used for secure remote management is TLS v1.2.

3.5 TOE FUNCTIONAL DIFFERENCES

There are no functional differences.

The TOE implements the following security functionality throughout all the models.

3.5.1 SECURITY AUDIT

The TOE's Audit security function supports audit record generation and review. The TOE provides date and time information that is used in audit timestamps. The Audit events generated by the TOE include:

- Establishment of a Trusted Path or Channel Session
- Failure to Establish a Trusted Path or Channel Session
- Termination of a Trusted Path or Channel Session
- Failure of Trusted Channel Functions
- Identification and Authentication
- Unsuccessful attempt to validate a certificate
- Changes to trust anchors in the TOE's trust store
- Any update attempts
- Result of the update attempt
- Management of TSF data
- Changes to Time

- Session termination for inactivity
- Power-on self tests verification
- Changes to audit server configuration
- Users locked out due to failed authentication attempts.

The TOE can store the generated audit data on itself, and it can be configured to send syslog events to a syslog server, using a TLS protected collection method. Logs are classified into various predefined categories. The logging categories help describe the content of the messages that they contain. Access to the logs is restricted to only Security Administrators, who are authorized to edit them, copy, or delete (clear) them. Audit records are protected from unauthorized modifications and deletions.

The TSF provides the capability to view audit data by using the Syslog tab in the local console. The log records the time, host name, facility, application, and "message" (the log details). The previous audit records are overwritten when the allocated space for these records reaches the threshold on a FIFO basis.

3.5.2 CRYPTOGRAPHIC SUPPORT

The TOE includes an OpenSSL library (Version 1.1.1k with Fedora Patches) that implements CAVP validated cryptographic algorithms for random bit generation, encryption/decryption, authentication, and integrity protection/verification. These algorithms are used to provide security for the TLS/HTTPs connections for secure management and secure connections to a syslog and authentication servers. TLS and HTTPs are also used to verify firmware updates. The cryptographic services provided by the TOE are described below:

Table 2 – TOE Cryptographic Protocols

Cryptographic Protocol	Use within the TOE
HTTPS/TLS (client)	Secure connection to syslog FCS_HTTPS_EXT.1, FCS_TLSC_EXT.1
HTTPS/TLS (server)	Peer connections to MAGNUM and remote management FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
AES	Provides encryption/decryption in support of the TLS protocol. FCS_COP.1.1/DataEncryption, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

DRBG	Deterministic random bit generation use to generate keys. FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, FCS_RBG_EXT.1
Secure hash	Used as part of digital signatures and firmware integrity checks. FCS_COP.1/Hash, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
HMAC	Provides keyed hashing services in support of TLS. FCS_COP.1/KeyedHash, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
EC-DH	Provides key establishment for TLS. FCS_CKM.2, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
ECDSA	Provides components for EC-DH key establishment. FCS_CKM.1, FCS_CKM.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
RSA	Provide key establishment, key generation and signature generation and verification (PKCS1_V1.5) in support of TLS. FCS_CKM.1, FCS_CKM.2, FCS_COP.1/SigGen, FCS_COP.1/SigVer, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

Each of these cryptographic algorithms have been validated for conformance to the requirements specified in their respective standards, as identified below and are part of the EXE Cryptographic Module.

Table 3 – CAVP Algorithm Testing References

Algorithm	Standard	CAVP Certificate #	Processors
AES 128/256-bit CBC, GCM, CTR	ISO 10116 (CBC and CTR) IOS 19772 (GCM)	A2573	Intel(R) Core (TM) i3-6102E
CTR DRBG using AES 256	ISO/IEC 18031:2011	A2573	Intel(R) Core (TM) i3-6102E
EC-DH	NIST SP 800-56A (key establishment)	A2573	Intel(R) Core (TM) i3-6102E

P-256, P-384, P-521			
ECDSA P-256, P-384, P-521	FIPS PUB 186-4 (key generation)	A2573	Intel(R) Core (TM) i3-6102E
HMAC-SHA-1/256/384	ISO/IEC 9797-2:2011	A2573	Intel(R) Core (TM) i3-6102E
SHA-1/256/384	ISO/IEC 10118-3:2004	A2573	Intel(R) Core (TM) i3-6102E
RSA 2048/3072/4096	FIPS PUB 186-4 (key generation and Digital Signature) ISO/IEC 9796-2 (digital signature)	A2573	Intel(R) Core (TM) i3-6102E

3.5.3 IDENTIFICATION AND AUTHENTICATION

All Administrators wanting to use TOE services are identified and authenticated prior to being allowed access to any of the services other than the display of the warning banner. ("Regular" EXE users do not access EXE directly; they control IP video switching through the EXE using a switch control system, such as Evertz'

Magnum. The switching of those IP video transport streams is outside the scope of the TOE.)

Once an Administrator attempts to access the management functionality of the TOE, the TOE prompts the Administrator for a username and password for password-based authentication. The identification and authentication credentials are confirmed against a local user database. Only after the Administrator presents the correct identification and authentication credentials will access to the TOE functionality be granted. If the user fails to provide the correct authentication credentials, the user will be locked out after a configurable threshold until the user is manually unlocked by an Administrator.

The TOE provides the capability to set password minimum length rules. This is to ensure the use of strong passwords in attempts to protect against brute force attacks. The TOE also accepts passwords

composed of a variety of characters to support complex password composition. During authentication, no indication is given of the characters composing the password.

Remote administrators are locked out after a configurable number of unsuccessful authentication attempts.

The EXE requires a password-protected serial connection to perform initial configuration of the system IP address(es). Once each address is established, administrators use IP connectivity for all further administrative actions, including configuration, operations, and monitoring.

The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS/HTTPS connections.

3.5.4 SECURITY MANAGEMENT

The TOE provides secure administrative services for management of general TOE configuration and the security functionality provided by the TOE. All TOE administration occurs either through a secure session or a local console connection. The TOE provides the ability to perform the following actions:

- Administer the TOE locally and remotely;
- Configure the access banner;
- Configure the session inactivity time before session termination or locking;
- Update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Specify the time limits of session inactivity;
- Ability to modify the IP address and the port of the remote syslog server;
- Generate Certificate Signing Requests, import and manage x509 certificates, delete/replace x509 certificates;
- Re-enable an Administrator account;
- Set the time which is used for timestamps.

All these management functions are restricted to Security Administrators who are authorized to administer the TOE via a local CLI and a remote web interface. Administrators are individuals who manage specific types of administrative tasks. The EXE implements role-based access control of these management functions to users that have been identified, authenticated, and authorized with the Security Administrator role.

Primary management is done using the Webeasy web-based interface using HTTPS. This provides a network administration console from which one can manage various identity services. These services include authentication, authorization, and reporting. All these services can be managed from the interface, which uses a menu-driven navigation system.

There is also a very simple serial-based connection (DB9) that provides a simple menu interface. This is used to configure the IP interface (IP address, etc.). It is password-protected, and is typically only used once, for initial set-up.

3.5.5 PROTECTION OF THE TSF

The TOE will terminate inactive sessions after an Administrator-configurable time period. Once a session has been terminated the TOE requires the user to re-authenticate to establish a new session. The TOE provides protection of TSF data (authentication data and cryptographic keys). In addition, the TOE internally maintains the date and time. This date and time are used as the time stamp that is applied to TOE generated audit records. The TOE also ensures firmware updates are from a reliable source. Finally

the TOE performs testing to verify correct operation.

An administrator initiates update processes from the web interface for all update installations. EXE automatically uses the RSA digital signature mechanism to confirm the integrity of the product before installing the update.

3.5.6 TOE ACCESS

Aside from the automatic Administrators session termination due to inactivity described above, the TOE also allows Administrators to terminate their own interactive session. Once a session has been terminated the TOE requires the user to re-authenticate to establish a new session.

The TOE will display an Administrator-specified banner on the web browser management interface prior to allowing any administrative access to the TOE.

3.5.7 TRUSTED PATH/CHANNELS

The TOE allows the establishment of a trusted channel

between a video control system (such as Evertz' Magnum) and EXE. The TOE also establishes a secure connection for sending syslog data to a syslog server using TLS.

The TOE uses HTTPS/TLS to provide a trusted path between itself and remote administrative users. The TOE does not implement any additional methods of remote administration. The remote administrative users are responsible for initiating the trusted path when they wish to communicate with the TOE.

3.6 ARCHITECTURAL DESCRIPTION.

The architectural description can be found in **Table 1 – TOE hardware models** above.

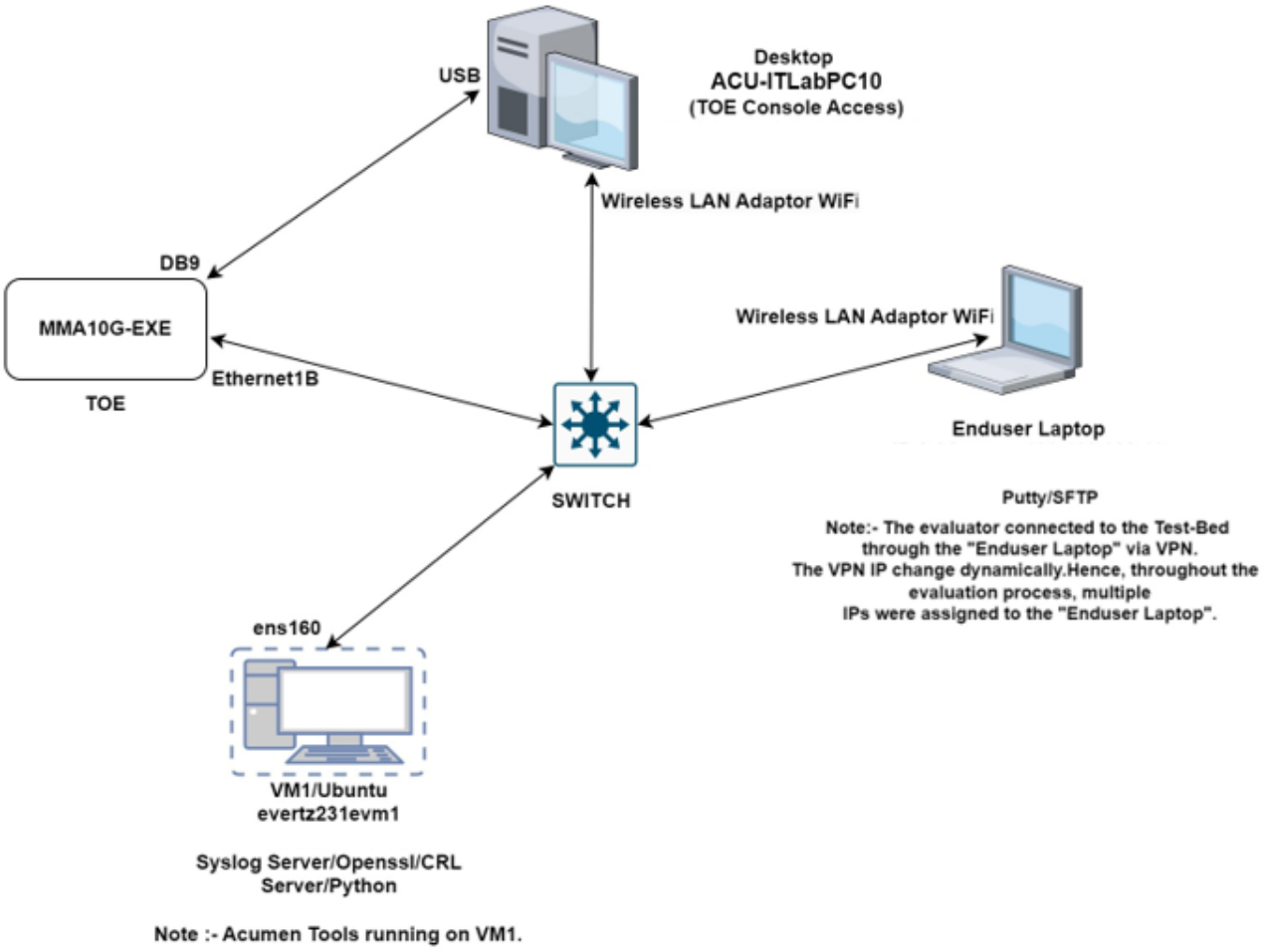
3.7 CONCLUSION.

Based on the equivalency rationale listed above, testing on only one model is sufficient. All other models listed above are included by equivalency. The following platform was tested end-to-end locally at Acumen Security:

- MMA10G-IPX-128 running EXE firmware Version 1.5

4 TEST BED DESCRIPTIONS

4.1 TEST BED



4.2 CONFIGURATION INFORMATION

The following table provides configuration information about each device in the test environment.

Device Name	OS	Version	Function	Protocols	Time	Tools (Version)
MMA10G-IPX-128	MMA10G-EXE	1.5	TOE	TLS/HTTPS	Manually set and verified	NA

VM(evertz231evm1)	Ubuntu	22.04.3 LTS	Syslog Server CRL Server Magnum Server	TLS,SSH	Manually set and verified	Tcpdump(4.99.1) Acumen-tools Openssl(3.0.2) Python
ACU-ITLabPC10	Windows 10 pro	22H2 build(19045.4291)	TOE Console Access	Serial, RDP	Manually set and verified	Putty(0.77) 64bit
Switch	Cisco IOS	NA	Provide Connectivity to TOE devices	IP	N/A	N/A
End User Laptop	Windows 10 pro	22H2 build(19045.4291)	Testing	SFTP,HTTPS	Manually set and verified	Wireshark(3.6.5) 64bit Putty

4.3 TEST TIME AND LOCATION

All testing was carried out remotely from Acumen Security offices located at 2400 Research Blvd Suite #395, Rockville, MD 20850.

Testing occurred from **February/2024 to July/2024.**

The TOE was in a physically protected, access controlled, designated test lab with no unattended entry/exit ways. At the start of each day, the test bed was verified to ensure that it was not compromised.

All evaluation documentation was always kept in a secure repository.

5 DETAILED TEST CASES (TSS AND AGD ACTIVITIES)

5.1 MANDATORY REQUIREMENTS

5.1.1 SECURITY AUDIT (FAU)

5.1.1.1 FAU_GEN.1 AUDIT DATA GENERATION

5.1.1.1.1 FAU_GEN.1 TSS

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.

Evaluator Findings:
The evaluator examined the TSS **FAU_GEN.1** and ensured that it identifies what information is logged to identify the relevant cryptographic key during generating/import, changing, or deleting. The relevant information is found in the following section(s): TOE Summary Specification **FAU_GEN.1** Upon investigation, the evaluator found that the TSS states that: **In the logs of Administrator actions which involve cryptographic keys (generating or deleting keys), the audit log will refer to the key as the "server private key".**

For distributed TOEs the evaluator shall examine the TSS and ensured that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components.

Evaluator Findings:
The TOE is not a distributed TOE; hence this assurance activity is not applicable.

The evaluator shall ensure that the 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 (as applicable to the overall TOE). The evaluator confirmed that all components defined as generating audit information for a particular SFR contributed to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component covered all the SFRs that it implements.

Evaluator Findings:
The TOE is not a distributed TOE; hence this assurance activity is not applicable.

Verdict:
PASS.

5.1.1.1.2 FAU_GEN.1 AGD

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).

Evaluator Findings:
The evaluator checked the AGD and ensured 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, was provided from the actual audit record). The relevant information is found in the following section(s): **"Audit Events"** Upon investigation, the evaluator found the table **'Audit Events Table'** contains a listing and description of each of the fields in generated audit records that contain the information required in FAU_GEN.1.2, as well as an example audit record. The evaluator next compared this list of events to the auditable events listed in the NDcPP and found that the table covers examples of all auditable events mentioned in the PP.

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.

Evaluator Findings:

The evaluator examined the AGD and verified that it identifies 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 first examined the entirety of AGD to determine what administrative commands are associated with each administrative activity. Upon investigation, the evaluator found that the following are applicable:

Administrative Activity	**Method (Command/GUI Configuration)**	**Section/s**
login and logout	**Logging to Local Console:** *Over a serial console port by using a 'Serial Connection Program' such as putty.exe, and login to the CLI using the username and password* **Logout of Local Console:**	• Login via Local Serial Connection • Terminating Serial Console Connection • Login via Web GUI • Terminating Web Session

	Enter 'X' until termination of the serial connection **Logging in to Web Interface:** • *Launch a web browser session* • *Enter 'https://<IP address of the EXE>"* • *Log in with username of the administrator and the password* **Logging out of Web Interface:** • *Click 'Logout' button on the top right corner*	
Resetting passwords	Change User Passwords: • *Login to the "**Management Web Application**"* • *Click "**Settings**" displayed at the bottom of the displayed page* • *Select "**Users**" tab* • *Select "**Edit**" to modify a user password.*	• User Management
Create CSR	• Login to the EXE **Management Web Application** • Click on "**General**" Tab from left side menu items • Click on "**Download**" button of "**CSR Regenerate And Download**" under Certificates section. EXE does not allow the configuration of CSR parameters; the following default parameters are used. These parameters will be customizable starting v1.7.- ○ Country Name: Canada ○ State or Province Name: Ontario ○ Locality Name: Burlington ○ Organization Name: Evertz Microsystems Ltd. ○ Organizational Unit Name: EXE ○ Common Name: Configured primary IP address of EXE	• Configure TLS Server, sub-section "Create Certificate Signing Request"

	○ Email Address: support@evertz.com	
Import Signed Server Certificate	• Login to the EXE **Management Web Application.** • Click "**General**" menu from Menus listed on left of the page. • Scroll down to "**Certificates**" section. • Click "**Choose File**" button of "**Certificate Upload**" segment and select the CA signed SSL certificate provided by your CA from your file system. • Click "**Upload**". • Wait for Upload success status to be displayed. • Reboot EXE.	• Configure TLS Server, sub-section "Upload SSL Certificate"
Import Trusted CA Certificate	• Login to the EXE **Management Web Application.** • Click "**General**" menu from Menus listed on left of the displayed index page. • Scroll down to "**Certificate**" section. • Click "**Choose File**" button of "**Certificate Chain Upload**" segment and select the trusted certificate chain provided by your CA from your file system. • Click "**Upload**". • A message informing the status of the upload will be displayed.	• Configure TLS Server, sub-section "Upload Certificate Chain"
Upgrading Firmware	• Login to the **Management Web Application** • Click "**Upgrade**" menu on top the displayed page	• Performing Secure Upgrade, sub-section "Upgrade"

		• Scroll to "**Image Settings**" Section • Find a slot which is empty. If None of the **Image Slots** are empty, click **Delete** button from a suitable Image slot • Click "**Choose File**" displayed in the Image Slot row, Select the image file to be upgraded to • Click "**Create**" button • Confirm the popup dialog • Wait for "**Processing**" status "**Message**" text to turn to "Image [N] created successfully using <filename>" • Image has been successfully upgraded into the slot location • Scroll up to "**Boot Image**" section and Select "**Next boot Image**" to the newly uploaded image slot • Click "**Reboot** button", wait for system to reboot in to the newly uploaded image	

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.

Evaluator Findings:

The evaluator examined each of the test cases and identified test cases exercised the above referenced functionality. The audit record associated with the configuration was captured. The following table identifies the test cases in which audit records for those configurations can be found.

Administrative Activity	**Method (Command/GUI Configuration)**	**Test Case(s)**
login and logout	**Logging to Local Console:** *Over a serial console port by using a 'Serial Connection Program' such as putty.exe, and login to the CLI using the username and password* **Logout of Local Console:** *Enter 'X' until termination of the serial connection* **Logging in to Web Interface:** • *Launch a web browser session* • *Enter 'https://<IP address of the EXE>"* • *Log in with username of the administrator and the password* **Logging out of Web Interface:** *Click 'Logout' button on the top right corner*	• FIA_UIA_EXT.1 Test #1 • FTA_SSL.4 Test #1 • FTA_SSL.4 Test #2
Resetting passwords	Change User Passwords: • *Login to the "**Management Web Application**"* • *Click "**Settings**" displayed at the bottom of the displayed page* • *Select "**Users**" tab* • *Select "**Edit**" to modify a user password.*	• FIA_PMG_EXT.1.1 Test #1
Create CSR	• Login to the EXE **Management Web Application** • Click on "**General**" Tab from left side menu items	• FIA_X509_EXT.3 Test #1

	• Click on "**Download**" button of "**CSR Regenerate And Download**" under Certificates section. EXE does not allow the configuration of CSR parameters; the following default parameters are used. These parameters will be customizable starting v1.7.- ○ Country Name: Canada ○ State or Province Name: Ontario ○ Locality Name: Burlington ○ Organization Name: Evertz Microsystems Ltd. ○ Organizational Unit Name: EXE ○ Common Name: Configured primary IP address of EXE ○ Email Address: support@evertz.com	
Import Signed Server Certificate	• Login to the EXE **Management Web Application.** • Click "**General**" menu from Menus listed on left of the page. • Scroll down to "**Certificates**" section. • Click "**Choose File**" button of "**Certificate Upload**" segment and select the CA signed SSL certificate provided by your CA from your file system. • Click "**Upload**". • Wait for Upload success status to be displayed. • Reboot EXE.	• FIA_X509_EXT.3 Test #2
Import Trusted CA Certificate	• Login to the EXE **Management Web Application.**	• FIA_X509_EXT.1.1/Rev Test #1a

	• Click "**General**" menu from Menus listed on left of the displayed index page. • Scroll down to "**Certificate**" section. • Click "**Choose File**" button of "**Certificate Chain Upload**" segment and select the trusted certificate chain provided by your CA from your file system. • Click "**Upload**". • A message informing the status of the upload will be displayed.	
Upgrading Firmware	• Login to the **Management Web Application** • Click "**Upgrade**" menu on top the displayed page • Scroll to "**Image Settings**" Section • Find a slot which is empty. If None of the **Image Slots** are empty, click **Delete** button from a suitable Image slot • Click "**Choose File**" displayed in the Image Slot row, Select the image file to be upgraded to • Click "**Create**" button • Confirm the popup dialog • Wait for "**Processing**" status "**Message**" text to turn to "Image [N] created successfully using <filename>" • Image has been successfully upgraded into the slot location	• FPT_TUD_EXT.1 Test #1

	• Scroll up to "**Boot Image**" section and Select "**Next boot Image**" to the newly uploaded image slot • Click "**Reboot** button", wait for system to reboot into the newly uploaded image	

Verdict:
PASS.

5.1.1.2 FAU_GEN.2 USER IDENTITY ASSOCIATION

5.1.1.2.1 TSS & AGD

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

5.1.1.3 FAU_STG_EXT.1 PROTECTED AUDIT EVENT STORAGE

5.1.1.3.1 FAU_STG_EXT.1 TSS

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.

Evaluator Findings:
The evaluator examined the TSS **FAU_STG_EXT.1** and ensured that it describes how the audit data is transferred to the external audit server, and how the trusted channel is provided. The relevant information is found in the following section(s): TOE Summary Specification **FAU_STG_EXT.1** Upon investigation, the evaluator found that the TSS states that: **Logs information is also sent to an external Syslog server via 'Syslog over TLS using TLS v1.2'. Logs are sent to the Syslog servers in real-time. For this to happen, an external syslog server should be configured (IP address/TCP Port number). A trusted certificate chain that is used to sign syslog server's certificate must be also uploaded to EXE.The [EXE CC Admin Guide] explains how to configure this connection. The trusted channel with the Syslog server is described in greater detail in the FCS_TLSC_EXT.1 description.**

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.

Evaluator Findings:
The evaluator examined the TSS **FAU_STG_EXT.1** and ensured 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 relevant information is found in the following section(s): TOE Summary Specification **FAU_STG_EXT.1** Upon investigation, the evaluator found that the TSS states that: **EXE stores audit logs internally. The internal logs are stored unencrypted, but they are only accessible (and then read-only) via the web browser, which can only be used by Administrators. Logs are initially written to messages file on /var/log/ directory and then moved to /nv/syslog/current when /var/log is full. The size limit for /var/log/ folder depends on the size of the memory used on each model. This folder can also contain files other than messages (syslog files), hence, the amount of audit logs that can be saved in the /var/log/ directory can vary. The current audit log is saved in the file name 'messages'. Once the current messages file reaches 60MB, it will be saved as messages.0 and a new messages file will be generated to capture the new audit logs. The full messages files will be written to messages.0, messages.1, and up to messages.10. As each messages.X file is created, it is archived and sent to the /nv/syslog/current/ directory. The /nv/syslog/current/ is in the hard disk and has a size limit of 880MB.** **The TOE overwrites previous audit records on a circular (FIFO) basis when both the volatile /var/log and persistent /nv/syslog/current storage space for audit is full.**

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.

Evaluator Findings:
The relevant information is found in the following section(s): TOE Summary Specification **FAU_STG_EXT.1** Upon investigation, the evaluator found that the TSS states that: **The TOE is a standalone TOE.**

The evaluator shall examine the TSS to ensure that for distributed TOEs it contains a list of TOE components that store audit data locally.

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

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.

Evaluator Findings:
The TOE is not a distributed TOE; hence this assurance activity is not applicable.

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.

Evaluator Findings:
The evaluator examined the TSS **FAU_STG_EXT.1** and ensured that it details the behaviour of the TOE when the storage space for audit data is full. The relevant information is found in the following section(s): TOE Summary Specification **FAU_STG_EXT.1** Upon investigation, the evaluator found that the TSS states that: **Logs are stored in /var/log. Logs are moved to /nv/syslog/current when /var/log is full. Information is also sent (using TLS 1.2) to an external Syslog server. The TOE overwrites previous audit records on a circular (FIFO) basis when the volatile /var/log and persistent /nv/syslog/current storage space for audit is full.**

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.

Evaluator Findings:
The relevant information is found in the following section(s): TOE Summary Specification **FAU_STG_EXT.1** Upon investigation, the evaluator found that the TSS states that: **EXE stores audit logs internally in real-time.** **Logs information is also sent to an external Syslog server via 'Syslog over TLS using TLS v1.2'. Logs are sent to the Syslog servers in real-time.**

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).

Evaluator Findings:
The TOE is not a distributed; TOE hence this assurance activity is not applicable.

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.

Evaluator Findings:
The TOE is not a distributed; TOE hence this assurance activity is not applicable.

Verdict:
PASS.

5.1.1.3.2 FAU_STG_EXT.1 AGD

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.

Evaluator Findings:
The evaluator examined the guidance documentation **"Offloading Audit Logs"** and ensured it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server, as well as configuration of the TOE needed to communicate with the audit server. The relevant information is found in the following section(s): **"Offloading Audit Logs"** Upon investigation, the evaluator found that the AGD states that: **System log messages can be sent to a remote audit server. The remote audit server must listen on TCP Port 6514 for TLSv1.2 connections, and its certificate chain must be trusted by EXE when Secure Mode is enabled. All audit events are simultaneously sent to the remote server and the local store. If this or any outgoing client connection is unintentionally broken, EXE will automatically reconnect within seconds.** **<u>Prerequisites</u>** • **A syslog server which supports secure TLS communication is up and running listening on TCP port 6514.** • **The syslog server supports TLS protocol version 1.2 and supports the ciphersuites listed in the section 2.4.6 above** **It also describes the steps on how to establish the trusted channel to the audit server. In addition, the syslog server requirements are also described.**

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.

Evaluator Findings:
The evaluator also examined the guidance documentation **"Offloading Audit Logs"**and determined that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. The relevant information is found in the following section(s): **"Offloading Audit Logs"** Upon investigation, the evaluator found that the AGD states that: **All audit events are simultaneously sent to the remote server and the local store. If this or any outgoing client connection is unintentionally broken, EXE will automatically reconnect within seconds.**

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.

Evaluator Findings:
The following information found under **"Viewing Audit Events via Web Interface"** section of the AGD: **The EXE can be operated as a standalone Network Device. EXE stores audit logs internally in real-time. The internal logs are stored unencrypted, but they are only accessible as a downloadable tar file.** **For local audit log storage, multiple log files are generated, each with a maximum capacity of approx. 60 MB. Once the current log file is full under "/var/log" path it is log-rotated., and simultaneously the old log-rotated logs are compressed and saved under a long-term storage location "/ssd/syslog/current" path. Compressed old log-rotated log files under the long-term storage are cleared based on a first-in-first-out basis with approximate maximum compressed logs of number 100. The audit logs will keep getting overwritten(log-rotated) with new files and audit log storage will never become full. This is the default behaviour and this cannot be modified by the administrators. In the CC evaluated configuration, the audit log path cannot be accessed by the recovery user through the console.** In the above, the AGD states that the behaviour when the local storage space is full cannot be modified by the administrators. The validator verified that this information corresponds to what is described in the TSS.

Verdict:
PASS.

5.1.2 CRYPTOGRAPHIC SUPPORT (FCS)

5.1.2.1 FCS_CKM.1 CRYPTOGRAPHIC KEY GENERATION

5.1.2.1.1 FCS_CKM.1 TSS

The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE.

Evaluator Findings:
The evaluator ensured that the TSS **FCS_CKM.1** identifies the key sizes supported by the TOE. The relevant information is found in the following section(s): TOE Summary Specification **FCS_CKM.1** Refer to below for evidence.

If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

Evaluator Findings:
The evaluator examined the TSS **FCS_CKM.1** and verified that it identifies the usage for each scheme. The relevant information is found in the following section(s): TOE Summary Specification **FCS_CKM.1** Upon investigation, the evaluator found that the TSS states that: **The TSF supports generation of 2048-bits, 3072-bits, and 4096-bits RSA keys for digital signatures in support of TLS sessions (FCS_TLSC_EXT.1 and FCS_TLSS_EXT.2) and the server certificate (FIA_X509_EXT.3).** **Generation of ECSA keys with NIST curves of P-256 or P-384 or P-521 are also used to generate ECDH components for key establishment in TLS sessions (FCS_TLSC_EXT.1 and FCS_TLSS_EXT.2).**

Verdict:
PASS.

5.1.2.1.2 FCS_CKM.1 AGD

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.

Evaluator Findings:
The evaluator found that the AGD **'Key Parameters'** states that the key parameters are non-configurable. The AGD states that: **EXE accepts 2048-bits, 3072-bits, and 4096-bits RSA keys from the TLS Clients and TLS Servers (with mutual authentication) but EXE only generates 2048-bit RSA keys during Certificate Signing Request generation. EXE does not allow or provide interfaces for the administrator to configure key generation parameters; Parameters are configured implicitly in accordance with the CC evaluation criteria.**

Verdict:
PASS.

5.1.2.2 FCS_CKM.2 CRYPTOGRAPHIC KEY ESTABLISHMENT

5.1.2.2.1 FCS_CKM.2 TSS **[TD0580]**

The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1.

Evaluator Findings:
The evaluator ensured 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 as shown in the table. 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.

Evaluator Findings:
The evaluator examined the TSS to verify that it identifies the usage for each scheme. The relevant information is found in the following section(s): TOE Summary Specification **FCS_CKM.2** Upon investigation, the evaluator found that the TSS states that: **The TOE acts as both sender and recipient for elliptic curve Diffie-Hellman key establishment schemes that meet the following:** • **NIST Special Publication (SP) 800-56A revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" – for FCS_TLSC_EXT.1 connections to the audit server and FCS_TLSS_EXT.2 connections to the MAGNUM server.** **or** • **RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specification Version 2.1". The TOE uses RSA-based key establishment for backwards compatibility for FCS_TLSC_EXT.1 connections to audit server and FCS_TLSS_EXT.2 connections to the MAGNUM server.**

Verdict:
PASS.

5.1.2.2.2 FCS_CKM.2 AGD

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

Evaluator Findings:
The evaluator found that the AGD **'Key Parameters'** states that the key parameters are non-configurable. The AGD states that: **EXE accepts 2048-bits, 3072-bits, and 4096-bits RSA keys from the TLS Clients and TLS Servers (with mutual authentication) but EXE only generates 2048-bit RSA keys during Certificate Signing Request generation. EXE does not allow or provide interfaces for the administrator to configure key generation parameters; Parameters are configured implicitly in accordance with the CC evaluation criteria.**

Verdict:
PASS.

5.1.2.3 FCS_CKM.4 CRYPTOGRAPHIC KEY DESTRUCTION

5.1.2.3.1 FCS_CKM.4 TSS

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 for2). 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.

Evaluator Findings:
The evaluator examined the TSS **FCS_CKM.4** to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations, and the destruction method used in each case. Upon investigation, the evaluator found that the TSS states that: **Cryptographic keys are destroyed by first overwriting the key file content with zeros. A read-verification is then performed to ensure that the entire content has really been changed to zeros and not any other values. If these steps fail, then the file will be overwritten again with zeros until the read-verify step succeeds. A sudden, unexpected power could disrupt zeroization and cause keys to not be zeroized. There are no other known circumstances where the TOE would not conform to these requirements.** The evaluator examined the section titled **TOE Summary Specification** in the Security Target to verify that the TSS description of keys and storage locations is consistent with the functions carried out by the TOE. Upon investigation, the evaluator found that the TSS states that: **The keys/CSPs used by the TOE, their storage location and format, and their associated zeroization method are as below:** • **EC Diffie-Hellman Keys** ○ **Storage location and method:** ***Plaintext in RAM*** ○ **Usage:** ***Key agreement and key establishment***

- **Zeroization:** ***Overwritten with zeroes when no longer needed.***
- **Firmware Update Key**
  - **Storage location and method:** ***Public key is stored in plaintext in the Flash disk. Private key is not stored or used on the TOE.***
  - **Usage:** ***Verification of firmware integrity when updating to new firmware versions using a SHA-256 hashed Public Key RSA signature.***
  - **Zeroization:** ***Public key in non-volatile storage (RAM) is automatically replaced once new firmware is booted. Public key, which is part of non-volatile firmware image is replaced with new firmware image when the new image is installed on top of the existing image slot. zeroize does not act on this file.***
- **HTTPS/TLS Server/Host Key**
  - **Storage location and method:** ***Plaintext in RAM.***
  - **Usage:** ***RSA and EC private key used in the HTTPS/TLS protocols***
  - **Zeroization:** ***During boot they get erased. When the client closes TLS session, the keys get erased.***
- **HTTPS/TLS session authentication key**
  - **Storage location and method:** ***Plaintext in RAM.***
  - **Usage:** ***HMAC SHA*-1, -256, or -384 key used for HTTPS/TLS session authentication.**
  - **Zeroization:** ***During boot they get erased. When the client closes TLS session, the keys get erased.***
- **HTTPS/TLS Session Encryption Key**
  - **Storage location and method:** ***Plaintext in RAM.***
  - **Usage:** ***AES (128, 256) key used for HTTPS/TLS session encryption***
  - **Zeroization:** ***During boot they get erased. When the client closes TLS session, the keys get erased.***
- **Locally Stored Passwords**
  - **Storage location and method:** ***SHA-256 Hashed in configuration file***
  - **Usage:** ***User Authentication***
  - **Zeroization:** **Temporary copy is created, modified, and replace the old file when no longer needed.**
- **Configuration Encryption Key**
  - **Storage location and method:** ***Plaintext in the Flash Disk***
  - **Usage:** ***Configuration Encryption***
  - **Zeroization:** **Temporary copy is created, modified, and replace the old file when no longer needed.**

To delete the plain-text keys stored on the non-volatile NOR flash storage, direct interface/access is provided to view or modify the contents of these files. The CLI provides Security Administrators with a menu item to destroy all CSPs, which would initiate key destruction.

No direct interface/access is provided to view or modify the contents of the keys stored in the volatile memory. The TLS session keys stored in RAM are automatically destroyed when the TLS session ends.

The DRBG state is zeroized using a single overwrite of zeros when the TSF is shutdown or restarted.

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).

Evaluator Findings:
This information is present in the TSS. Refer to the evidence above for details.

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.

Evaluator Findings:
The evaluator checked the TSS **FCS_CKM.4** to ensure that the interface supports the selection(s) and description in the TSS. In the case of non-volatile memory, the evaluator examined the relevant interface description for each media type on which plaintext keys are stored. Upon investigation, the evaluator found that the TSS states that: **To delete the plain-text keys stored on the non-volatile NOR flash storage, direct interface/access is provided to view or modify the contents of these files. The CLI provides Security Administrators with a menu item to destroy all CSPs, which would initiate key destruction.** **No direct interface/access is provided to view or modify the contents of the keys stored in the volatile memory. The TLS session keys stored on Flash are automatically destroyed when the TLS session ends.** **The DRBG state is zeroized using a single overwrite of zeros when the TSF is shutdown or restarted.** **The above destruction methods are followed in all configurations and circumstances.**

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.

Evaluator Findings:
Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator checked that the TSS **FCS_CKM.4** 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. Upon investigation, the evaluator found that the TSS states that: **Cryptographic keys are destroyed by first overwriting the key file content with zeros. A read-verification is then performed to ensure that the entire content has really been changed to zeros and not any other values. If these steps fail, then the file will be overwritten again with zeros until the read-verify step succeeds. A sudden, unexpected power could disrupt zeroization and cause keys to not be zeroized. There are no other known circumstances where the TOE would not conform to these requirements.** **The keys/CSPs used by the TOE, their storage location and format, and their associated zeroization method are as below:** • **Locally Stored Passwords** ○ **Storage location and method: SHA-256 Hashed in configuration file** ○ **Usage: User Authentication** ○ **Zeroization: Temporary copy is created, modified, and replace the old file when no longer needed.**

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.

Evaluator Findings:
The evaluator checked that the TSS **FCS_CKM.4** identifies any configurations or circumstances that may not conform to the key destruction requirement. Upon investigation, the evaluator found that the TSS states that: **A sudden, unexpected power could disrupt zeroization and cause keys to not be zeroized. There are no other known circumstances where the TOE would not conform to these requirements.**

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.

Evaluator Findings:
The evaluator verified that ST does not specify the use of ‘a value that does not contain any CSP’ to overwrite keys.

Verdict:
PASS.

5.1.2.3.2 FCS_CKM.4 AGD

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).

Evaluator Findings:
The evaluator checked that the guidance documentation, section **“Zeroing Crypto Material"** 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.

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).

Evaluator Findings:
The evaluator checked that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer. The relevant information is found in the following section: **“Zeroing Crypto Material"**

Upon investigation, the evaluator found that the AGD states that:
Once Factory Reset command is executed successfully all sensitive key material and crypto specific data will be disposed PERMANANTLY during the reboot. The deletion is a straight-forward process and should not result in any delays. If the reboot process gets interrupted (due to power failure), the keys might not get permanently deleted. In such scenarios, the above steps will have to be repeated.

Verdict:
PASS.

5.1.2.4 FCS_COP.1/DATAENCRYPTION CRYPTOGRAPHIC OPERATION (AES DATA ENCRYPTION/DECRYPTION)

5.1.2.4.1 FCS_COP.1/DATAENCRYPTION TSS

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.

Evaluator Findings:
The evaluator examined the TSS **FCS_COP.1/DataEncryption** to ensure it identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption. Upon investigation, the evaluator found that the TSS states that: **The TOE provides AES encryption/decryption in CBC, CTR, or GCM mode with 128- and 256-bit keys.**

Verdict:
PASS.

5.1.2.4.2 FCS_COP.1/DATAENCRYPTION AGD

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.

Evaluator Findings:
The evaluator verified that the AGD guidance states that the data encryption/decryption modes and key sizes are non-configurable. The relevant information is found in the following section(s): **"Data Encryption/ Decryption Modes"** Upon investigation, the evaluator found that the AGD states that: **EXE only supports AES encryption and decryption in CBC and GCM modes with key sizes 128 and 256 for TLS. AES_CTR_384 is used for Random Bit Generation. All these modes and key sizes are supported by default and EXE does not allow or provide interfaces for the administrators to configure data encryption and decryption parameters. Parameters are hard coded implicitly in accordance with the CC evaluation criteria.**

Verdict:
PASS.

5.1.2.5 FCS_COP.1/SIGGEN CRYPTOGRAPHIC OPERATION (SIGNATURE GENERATION AND VERIFICATION)

5.1.2.5.1 FCS_COP.1/SIGGEN TSS

The evaluator shall examine the TSS to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services.

Evaluator Findings:
The evaluator examined the TSS **FCS_COP.1/SigGen** to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services. Upon investigation, the evaluator found that the TSS states that: **The TOE supports signature generation and verification with RSA (2048-bits, 3072- bits, and 4096-bits) with SHA-1/256/384 in accordance with FIPS PUB 186-4.** **These signatures support TLS authentication and firmware verification. The TOE's server certificate is always 2048-bits.**

Verdict:
PASS.

5.1.2.5.2 FCS_COP.1/SIGGEN AGD

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.

Evaluator Findings:
The evaluator verified that the AGD guidance states that the selected cryptographic algorithm and key size defined in the Security Target supported by the TOE for signature services are non-configurable. The relevant information is found in the following section(s): **"Cipher Suites"** and **"Key Parameters"** Upon investigation, the evaluator found that the AGD states that: **EXE does not allow or provide interfaces for the administrator to configure/enable/disable cipher suits. Rather EXE by default supports the following cipher-suits in compliance with CC evaluation criteria implicitly. No configuration is needed or possible in both cipher suits selection and RNG.** **EXE does not allow or provide interfaces for the administrator to configure key generation parameters; Parameters are hard coded implicitly in accordance with the CC evaluation criteria.**

Verdict:
PASS.

5.1.2.6 FCS_COP.1/HASH CRYPTOGRAPHIC OPERATION (HASH ALGORITHM)

5.1.2.6.1 FCS_COP.1/HASH TSS

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.

Evaluator Findings:
The evaluator checked that the association of the hash function with other TSF cryptographic functions is documented in the TSS. The relevant information is found in the following section(s): TOE Summary Specification **FCS_COP.1/Hash** Upon investigation, the evaluator found that the TSS states that: **The TOE implements hashing in byte-oriented mode. The TOE provides cryptographic hashing services in support of TLS for SHA-1, SHA-256 and SHA-384. SHA-256 is used in firmware integrity checks during power-on-self-tests and upgrades. The locally stored passwords are salted using SHA-256. Key generation is performed using SHA-256 as specified in NIST SP 800-90 DRBG.**

Verdict:
PASS.

5.1.2.6.2 FCS_COP.1/HASH AGD

The evaluator checks the AGD documents to determine that any configuration that is required to configure the required hash sizes is present.

Evaluator Findings:
The evaluator checked the AGD document to determine that any configuration that is required to configure the required hash sizes is present. The relevant information is found in the following section(s): **"Hash and Keyed-Hash Algorithms"** Upon investigation, the evaluator found that the AGD states that: **EXE does not allow or provide interfaces for the administrator to configure Hash or Keyed Hash algorithm parameters; Parameters are configured implicitly in accordance with the CC evaluation criteria. By default, EXE supports SHA-1, SHA-256, SHA-384 hash algorithms and HMAC-SHA1 with 160-bit key, HMAC-SHA256 with 256-bit key, HMAC-SHA384 384-bit key keyed hash algorithms.**

Verdict:
PASS.

5.1.2.7 FCS_COP.1/KEYEDHASH CRYPTOGRAPHIC OPERATION (KEYED HASH ALGORITHM)

5.1.2.7.1 FCS_COP.1/KEYEDHASH TSS

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.

Evaluator Findings:
The evaluator examined the TSS **FCS_COP.1/KeyedHash** 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. Upon investigation, the evaluator found that the TSS states that: **The following keyed-hash message authentication are used by EXE:** • **HMAC-SHA-1 with 160-bit keys, message digest size of 160 bits and 160-bits message block size,** • **HMAC-SHA-256 with 256-bit keys, message digest sizes of 256 bits, and block size of 512 bits, and** • **HMAC-SHA-384 with 384-bit keys, message digest sizes of 384 bits, and block size of 1024 bits.**

Verdict:
PASS.

5.1.2.7.2 FCS_COP.1/KEYEDHASH AGD

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.

Evaluator Findings:
The evaluator verified that the AGD guidance states that 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 are non-configurable. The relevant information is found in the following section(s): **"Hash and Keyed-Hash Algorithms"** Upon investigation, the evaluator found that the AGD states that: **EXE does not allow or provide interfaces for the administrator to configure Hash or Keyed Hash algorithm parameters; Parameters are configured implicitly in accordance with the CC evaluation criteria. By default, EXE supports SHA-1, SHA-256, SHA-384 hash algorithms and HMAC-SHA1 with 160-bit key, HMAC-SHA256 with 256-bit key, HMAC-SHA384 384-bit key keyed hash algorithms**.

Verdict:
PASS.

5.1.2.8 FCS_RBG_EXT.1 EXTENDED: CRYPTOGRAPHIC OPERATION (RANDOM BIT GENERATION)

5.1.2.8.1 FCS_RBG_EXT.1 TSS

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.

Evaluator Findings:
The evaluator examined the TSS **FCS_RBG_EXT.1** and determined 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. Upon investigation, the evaluator found that the TSS states that: **The TOE implements a DRBG in accordance with ISO/IEC 18031:2011 using a CTR DRBG with AES. The TSF seed the CTR_DRBG using 384-bits of data that contains at least 359 bits of entropy. The TSF gathers and pools entropy from two software-based noise sources: haveged and the Linux kernel provided entropy.** **The entropy sources are discussed in greater detail in the Entropy documentation.**

Verdict:
PASS.

5.1.2.8.2 FCS_RBG_EXT.1 AGD

The evaluator shall confirm that the guidance documentation contains appropriate instructions for configuring the RNG functionality.

Evaluator Findings:
The evaluator confirmed that the guidance documentation, section **"Data Encryption/Decryption Modes"** states that the RNG functionality is non-configurable. Upon investigation, the evaluator found that the AGD states that: **AES_CTR_384 is used for Random Bit Generation. All these modes and key sizes are supported by default and EXE does not allow or provide interfaces for the administrators to configure data encryption and decryption parameters. Parameters are hard coded implicitly in accordance with the CC evaluation criteria.**

Verdict:
PASS.

5.1.3 IDENTIFICATION AND AUTHENTICATION (FIA)

5.1.3.1 FIA_AFL.1 AUTHENTICATION FAILURE MANAGEMENT

5.1.3.1.1 FIA_AFL.1 TSS

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.

Evaluator Findings:
The evaluator examined the section titled **TOE Summary Specification** in the Security Target to verify that the TSS contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked; the method by which the remote administrator is prevented from successfully logging on to the TOE; and the actions necessary to restore this ability. Upon investigation, the evaluator found that the TSS states that: **An administrator can configure the number of unsuccessful attempts a remote administrator can make before a lock-out occurs. The attempts can range between 3 and 20. The default number of attempts is 10.** **If the user enters an incorrect password the configured number of times, the user is locked out and they cannot login through any remote interface on the TOE. The username will show the Lockout enabled on the settings->Users page on the web interface. Users must have an administrator unlock their account before they can regain access. Administrators can also have a different administrator unlock their account.**

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).

Evaluator Findings:
The evaluator examined the section titled **TOE Summary Specification** in the Security Target to verify that the TSS ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available. Upon investigation, the evaluator found that the TSS states that: **Lockouts are not enforced on the TOE's console interface. This ensures that authentication failures cannot lead to a situation where no administrator access is available.**

Verdict:
PASS.

5.1.3.1.2 FIA_AFL.1 AGD

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.

Evaluator Findings:
The evaluator examined the section titled **"Limit Login Attempts"** in the AGD to verify that it provides instructions for configuring the number of successive unsuccessful authentication attempts and time period, and that the process of allowing the remote administrator to once again successfully log on is described for each "action" specified. Upon investigation, the evaluator found that the AGD states that; **Steps** 1. **Login to the EXE Management Web Application**

2. **Click "Settings" button at the bottom right of the displayed index page**
3. **Click "Login" tab at the displayed Settings page**
4. **Scroll down to Login segment at the bottom of the Settings page**
5. **Set "Max Failed Login Attempts" to an acceptable value between "3" and "20"**
6. **Click "Apply" button**

In addition, section **'User Management'** in the AGD explains that a user is locked out when the above limit is exceeded and provides the steps to re-enable the locked-out users.

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.

Evaluator Findings:

The evaluator examined the section titled "**Limit Login Attempts"** in the AGD to verify 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. Upon investigation, the evaluator found that the AGD states that;

Above limit login attempt is applicable for WebGUI session. It is not applicable for local console sessions.

This ensures that authentication failures cannot lead to a situation where no administrator access is available.

Verdict:
PASS.

5.1.3.2 FIA_PMG_EXT.1 PASSWORD MANAGEMENT

5.1.3.2.1 FIA_PMG_EXT.1 TSS **[TD0792]**

The evaluator shall check that the TSS lists the supported special character(s) for the composition of administrator passwords.

Evaluator Findings:

The evaluator examined the section titled **TOE Summary Specification** in the Security Target to verify that the TSS lists the supported special character(s) for the composition of administrator passwords.

Upon investigation, the evaluator found that the TSS states that:

EXE enforces that passwords must meet minimum requirements such as length, mix of number, lower/upper case letters, and the following special characters "!"; "@"; "#"; "$"; "%"; "^"; "&"; "*"; "("; ")";"~"; "`"; "_"; "-"; "+"; "="; "{"; "["; "}"; "]"; "|"; "\"; ":"; ";"; ["]; ['"]; "<"; ","; ">"; "."; "?"; "/"; [space]. No common dictionary

words are allowed. At least two characters from each category are required (upper case letter, lower case letter, number special character).

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

Evaluator Findings:
The evaluated examined the TSS, section **FIA_PMG_EXT.1** and verified that the minimum_password_length parameter is configurable by a Security Administrator. Upon investigation, the evaluator found that the TSS states that: **Passwords must be at least a minimum length settable by the administrator and support 15 to 20 characters.**

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.

Evaluator Findings:
The evaluated examined the TSS, section **FIA_PMG_EXT.1** and verified that the range of values supported for minimum_password_length parameter is listed. Upon investigation, the evaluator found that the TSS states that: **Passwords must be at least a minimum length settable by the administrator and support 15 to 20 characters.**

Verdict:
PASS.

5.1.3.2.2 FIA_PMG_EXT.1 AGD

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.

Evaluator Findings:
The evaluator examined the section titled "**Secure Password"** in the AGD to verify that it identifies the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords and provides instructions on setting the minimum password length and describes the valid minimum password lengths supported. Upon investigation, the evaluator found that the AGD states that: **1. Login to the EXE Management Web Application.** **2. Click "Settings" button at the bottom right of the displayed index page.** **3. Click "Login" tab at the displayed Settings page.** a **Under "Password" section select "Password Strength" to "Strong".**

4. **Click "Apply" button.**

Once the above choice is made, EXE mandates following in terms of password requirement,

a) Passwords shall be able to be composed of any combination of upper- and lower-case letters, numbers, and the following special characters: ["!", "@", "#", "$", "%", "^", "&", "*", "(", ")", ["~", "`", "_", "-", "+", "=", "{", "[", "}", "]", "|", "\", ":", ";", (”), (’), "<", ",", ">", ".", "?", "/", (space)]];

b) Minimum password length is set to 15 characters by default.

To configure minimum password length between 15 to 20 characters,

a) Click on "Customization" Tab.

b) Enter the desired password length in the field for "minimum length" as shown in below image.

c) Click on "Apply" tab to finalize changes.

Verdict:
PASS.

5.1.3.3 FIA_UIA_EXT.1 USER IDENTIFICATION AND AUTHENTICATION

5.1.3.3.1 FIA_UIA_EXT.1 TSS

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".

Evaluator Findings:
The evaluator examined the section titled **TOE Summary Specification** in the Security Target to verify that the TSS describes the logon process for each logon method supported for the product. Upon investigation, the evaluator found that the TSS states that: **Administrators can log on via the web interface using HTTPS or locally on the serial port. A username and password is required to authenticate the administrator for both methods. The Security Administrator is considered authenticated if the username and password match the stored credential values.**

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.

Evaluator Findings:
The evaluator examined the TSS, section **FIA_UIA_EXT.1** of the ST and found that it describes which actions are allowed before administrator identification and authentication. Upon investigation, the evaluator found that the TSS states that: **Prior to successful identification and authentication on all interfaces, the TSF displays the TOE access banner specified in FTA_TAB.1. Users must acknowledge the warning banner before they can login to the system.**

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.

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

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.

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

Verdict:
PASS.

5.1.3.3.2 FIA_UIA_EXT.1 AGD

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.

Evaluator Findings:
The evaluator examined the guidance documentation, section **"Initial Configuration"** and determined that all necessary preparatory steps to logging in are described. This section describes all the prerequisites for each type of login method (Console and WebGUI) necessary for admins to administer the EXE locally and remotely.

Verdict:
PASS.

5.1.3.4 FIA_UAU_EXT.2 PASSWORD-BASED AUTHENTICATION MECHANISM

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.

5.1.3.5 FIA_UAU.7 PROTECTED AUTHENTICATION FEEDBACK

5.1.3.5.1 FIA_UAU.7 TSS

None.

5.1.3.5.2 FIA_UAU.7 AGD

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.

Evaluator Findings:
The evaluator examined the guidance documentation to determine that any necessary preparatory steps to ensure authentication data is not revealed while entering for each local login allowed. Upon investigation, the evaluator found that the AGD, section **"Accessing the EXE"** states that: **No Configuration is required to obscure the password.**

Verdict:
PASS.

5.1.4 SECURITY MANAGEMENT (FMT)

5.1.4.1 FMT_MOF.1/MANUALUPDATE

5.1.4.1.1 FMT_MOF.1/MANUALUPDATE TSS

For distributed TOEs see chapter 2.4.1.1. There are no specific requirements for non-distributed TOEs.

Evaluator Findings:
The TOE is not a distributed TOE and there are no specific requirements for non-distributed TOEs; Hence, this assurance activity is not applicable.

Verdict:
PASS.

5.1.4.1.2 FMT_MOF.1/MANUALUPDATE AGD

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).

Evaluator Findings:
The evaluator examined the guidance documentation and determined that any necessary steps to perform manual update are described. The guidance documentation also provides warnings regarding functions that may cease to operate during the update (if applicable). Upon investigation, the evaluator found that the **steps 1-11** in the section **"Performing Secure Upgrade"** describe the process of manually updating the software on the TOE.

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).

Evaluator Findings:
The TOE is not a distributed TOE; Hence, this assurance activity is not applicable.

Verdict:
PASS.

5.1.4.2 FMT_MTD.1/COREDATA MANAGEMENT OF TSF DATA

5.1.4.2.1 FMT_MTD.1/COREDATA TSS

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.

Evaluator Findings:
The relevant information is found in the following section(s): TOE Summary Specification **FMT_MTD.1/CoreData** The evaluator confirmed that the TSS details that the TSS identifies administrative functions that are accessible through an interface prior to administrator log-in. Upon investigation, the evaluator found that the TSS states that: **No administrative functionality is available prior to login. The TSF displays a warning banner prior to user authentication.** The evaluator examined the section titled **TOE Summary Specification** in the Security Target to verify that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users. Upon investigation, the evaluator found that the TSS states that: **The TSF implements the Security Administrator role to authorized administrators of the TOE. The TSF allows the Security Administrators to administer the TSF via a local CLI and a remote web interface. The TSF implements role-based access control of these management functions to users that have been identified, authenticated, and authorized with the Security Administrator role.** **When a user account is created (by administrator), it must be assigned with a role that specifies the privileges the account will have. The administrator can choose to assign an existing role with pre-defined privileges or create a new role with customized privileges.** **The (non-administrative) User has no direct access or control over EXE; a (non-administrative) User may only access an EXE card through MAGNUM. The (non-administrative) User can only view configurations.**

The administrative interfaces provided by the TSF do not allow any of these functions to be accessed by unauthenticated or unauthorized 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.

Evaluator Findings:

The evaluator examined the TSS and verified that it contains sufficient information to describe how the ability to manage the TOE's trust store is restricted.

The relevant information is found in the following section(s): TOE Summary Specification **FMT_MTD.1/CoreData**

Upon investigation, the evaluator found that the TSS states that:

The Web interface and local console allow the Security Administrator to perform the following TSF management functions:

- **Reset certificates.**
- **Import certificates.**
- **Import Trusted CA certificate.**
- **Delete (Replace) x509 certificates in the trust store;**
- **Create/Download a certificate signing request (CSR).**

The TOE maintains a trust store where the TOE's certificate is stored. Only Security Administrators have access to the trust store. Security Administrators can upload a certificate chain. Uploading the certificate chain replaces the previously installed certificate chain.

Verdict:
PASS.

5.1.4.2.2 FMT_MTD.1/COREDATA AGD

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 c PP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

Evaluator Findings:

The evaluator reviewed the guidance documentation and determined 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.

The relevant information is found in the following section(s): **"Secure Configuration"**

Upon investigation, the evaluator found that the AGD includes configurations of the following:

- **Configure Secure Mode**
- **Verify Power-On Self-Tests**
- **Verify Secure Mode Banners**
- **FIPS Mode**
- **Self-Test**
- **Cipher Suites**
- **Data Encryption/ Decryption Modes**
- **Key Parameters**
- **Hash and Keyed-Hash Algorithms**
- **Configure Access Controls**
  - **Unauthorized Access Prevention**
  - **Secure Passwords**
  - **Set Session Timeout**
  - **Configure Session Handling**
  - **Limit Login Attempts**
  - **Configure Secure Access Banner**
  - **Disable REST API**
  - **Disable SNMP**
  - **Disable NTP**
  - **Disable LLDP streaming**
  - **Disable LDAP**
- **Configure TLS Server**
  - **Download Certificate Signing Request**
  - **Signing the CSR using a Public or Organizational Certificate Authority**
  - **Upload Certificate Chain**
  - **Upload SSL Certificate**
- **Configure TLS Client**

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.

Evaluator Findings:

The evaluator reviewed the guidance documentation to determine that it provides sufficient information for the administrator to configure and maintain the trust store in a secure way. The evaluator also determined that it provides sufficient information for the administrator to securely load CA certificates into the trust store and explains how to designate a CA certificate a trust anchor.

The relevant information is found in the following section(s): **'Configure TLS Server' and 'Configure TLS Client'**

Upon investigation, the evaluator found that the AGD states required steps under above sections.

Verdict:
PASS.

5.1.4.3 FMT_SMF.1 SPECIFICATION OF MANAGEMENT FUNCTIONS

5.1.4.3.1 FMT_SMF.1 TSS (CONTAINING ALSO REQUIREMENTS ON GUIDANCE DOCUMENTATION AND TESTS)

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.

Evaluator Findings:
The evaluator examined the TSS of the ST, the Guidance Documentation, and the TOE as observed during all other testing and confirmed 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).

Evaluator Findings:
The evaluator confirmed that the TSS **FMT_SMF.1** details which security management functions are available through which interface(s) (local administration interface, remote administration interface). The relevant information is found in the following section(s): TOE Summary Specification **FMT_SMF.1** Upon investigation, the evaluator found that the TSS states that: **The web interface allows the Security Administrator to perform the following TSF management functions:** • **Edit login banner;** • **Create certificate signing request CSR, download a CSR;** • **Zeroize all Critical Security Parameters (CSP);** • **Import certificates;** • **Import Trusted CA certificate;** • **Delete (Replace) x509 certificates in the trust store;** • **Configure webGUI and console menu system timeout;** • **Verify/Install Firmware Updates;** • **View/Edit settings for sending audit data to the Syslog Server;** • **View/Edit authentication failure parameters;** • **Unlock a locked user after the login failure threshold is exceeded;** **The following can only be performed from the console interfaces:** • **Configure EXE date and time;** • **Control port IP configuration;** • **Reset certificates.**

The evaluator shall examine the TSS and Guidance Documentation to verify they both describe the local administrative interface.

Evaluator Findings:
The evaluator also found that the TSS section "**FMT_SMF.1"** describes the local administrative interface. Upon investigation, the evaluator found that the TSS states that: **Administrators can administer EXE locally through serial port connection. A console menu can be used to perform configurations tasks such as setting IP/system time/system reboot, etc.** The evaluator also found that the AGD section "**Accessing The EXE"** describes the local administrative interface. Upon investigation, the evaluator found that the AGD states that: **Administrators can administer EXE locally through serial port connection. A console menu can be used to perform configurations tasks such as setting IP/system time/system reboot, etc.**

The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

Evaluator Findings:
Regarding having appropriate warnings for the administrator to ensure the interface is local, the evaluator examined the admin guide and the TSS and verified that only two management interfaces described in the document as applicable to the TOE are **WebGUI** and the **Serial Console**. While a graphical user interface is being used for the remote management of the device via HTTPS, the local serial console uses a CLI. The common differences in these two types of interfaces are sufficient for security administrators to distinguish between the two and ensure that the Serial Interface connection is local. In addition, Admin Guide section "**Accessing the EXE**" describes the steps to ensure that the Security Administrators have successfully established a Serial Connection.

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.

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

Verdict:
PASS.

5.1.4.3.2 FMT_SMF.1 AGD

See section 2.4.4.1.

Evaluator Findings:
See section 5.1.4.3.1 of this document for AGD activities.

Verdict:
PASS.

5.1.4.4 FMT_SMR.2 RESTRICTIONS ON SECURITY ROLES

5.1.4.4.1 FMT_SMR.2 TSS

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.

Evaluator Findings:
The evaluator examined the TSS and determined that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE. The relevant information is found in the following section(s): TOE Summary Specification **FMT_SMR.2** Upon investigation, the evaluator found that the TSS states that: **The TSF implements the Security Administrator role to authorized administrators of the TOE. The TSF allows the Security Administrators to administer the TSF via a local CLI and a remote web interface. The TSF implements role-based access control of these management functions to users that have been identified, authenticated, and authorized with the Security Administrator role.** **When a user account is created (by administrator), it must be assigned with a role that specifies the privileges the account will have. The administrator can choose to assign an existing role with pre-defined privileges (administrative role, role with read-write user privileges, and role with read-only privileges) or create a new role with customized privileges.** **Administrators can administer EXE locally through serial port connection. A console menu can be used to perform configurations tasks such as setting IP/system time/system reboot, etc.** **Administrators can administer EXE remotely through its web interface, which runs on HTTPS. The web interface supports a broader set of configuration settings that include configurations for certificate imports, syslog server, route mapping, etc.** **The CLI allow the Security Administrator to perform the following TSF management functions on cryptographic keys:** • **TLS Key Generation (TLS keys are automatically generated when creating a CSR)** • **TLS Key Reset/Replacement (when a CSR is generated, previous TLS key will be deleted and replaced by the new key. The TLS keys cannot be imported from outside the TOE. The administrators cannot delete TLS keys manually).** **The administrative interfaces provided by the TSF do not allow any of these functions to be accessed by unauthenticated or unauthorized users.**

Verdict:
PASS.

5.1.4.4.2 FMT_SMR.2 AGD

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.

Evaluator Findings:
The evaluator reviewed the AGD and ensured 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. The relevant information is found in the following section: **Accessing the EXE**

Verdict:
PASS.

5.1.5 PROTECTION OF THE TSF (FPT)

5.1.5.1 FPT_SKP_EXT.1 PROTECTION OF TSF DATA (FOR READING OF ALL PRE-SHARED, SYMMETRIC AND PRIVATE KEYS)

5.1.5.1.1 FPT_SKP_EXT.1 TSS

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.

Evaluator Findings:
The evaluator examined the TSS and determined 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 describes how they are protected/obscured. The relevant information is found in the following section(s): TOE Summary Specification **FPT_SKP_EXT.1** and **in FCS_CKM.4** Upon investigation, the evaluator found that the TSS states that: **The TSF stores cryptographic keys in a directory. As there is no command line access, users cannot gain any direct access to these files.** **Information regarding the storage locations, usage, and method of storage of the cryptographic keys described in FCS_CKM.4.**

The evaluator reviewed **FCS_CKM.4** in the TSS and verified that all the information is present to cover the requirement. All keys are stored in plaintext except for the locally stored password file, which is protected by a SHA-256 hash.

Verdict:
PASS.

5.1.5.2 FPT_APW_EXT.1 PROTECTION OF ADMINISTRATOR PASSWORDS

5.1.5.2.1 FPT_APW_EXT.1 TSS

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.

Evaluator Findings:

The evaluator examined 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 relevant information is found in the following section(s): TOE Summary Specification
FPT_APW_EXT.1

Upon investigation, the evaluator found that the TSS states that:

The TSF does not store plaintext passwords. Passwords are hashed using SHA-256 and stored in a secure location which is not accessible to users. Secure (one-way) hash functions ensure that it's computationally impossible to recover a plaintext from its hashed value.

Verdict:
PASS.

5.1.5.3 FPT_TST_EXT.1 TSF TESTING

5.1.5.3.1 FPT_TST_EXT.1 TSS

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).

Evaluator Findings:

The evaluator examined the TSS and ensured that it details the self-tests that are run by the TS and that this description includes an outline of what the tests are actually.

The relevant information is found in the following section(s): TOE Summary Specification
FPT_TST_EXT.1

Upon investigation, the evaluator found that the TSS states that:

The TSF performs the following hardware self-tests at power-on:

- **firmware integrity check that compares the SHA256 checksum of the loaded firmware with a permanently stored hash value;**

The TSF enables FIPS mode on the OpenSSL library when Secure Mode is configured. Upon enabling FIPS mode the algorithm self-tests required by FIPS are performed. The OpenSSL library self-tests include:

- **Cryptographic library tests:**
  - **SHA-256 KAT.**
  - **HMAC-SHA-256 KAT.**
  - **AES 128 GCM Encrypt and Decrypt KAT.**
  - **RSA 4096 SHA-256 Sign and Verify KAT.**
  - **ECDSA Pairwise Consistency Test.**
  - **DRBG AES-CTR-256 KAT (invoking the instantiate, reseed, and generate functions).**

The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

Evaluator Findings:
The evaluator examined the section titled **TOE Summary Specification** in the Security Target to verify that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly. Upon investigation, the evaluator found that the TSS states that: **If any of the other checks fail, the TOE will fail to boot and an error will be displayed. Administrators are instructed to contact Evertz service department for repair if the failure does not clear on reboot. These self-tests ensure the TOE software has the correct image and that cryptographic functions are performing appropriately. If failures are seen by the Administrator, they should be immediately corrected.**

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.

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

Verdict:
PASS.

5.1.5.3.2 FPT_TST_EXT.1 AGD

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.

Evaluator Findings:
The evaluator also ensured 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 correspond to those described in the TSS. The relevant information is found in the following section(s): **Verify Power-On Self-Test** Upon investigation, the evaluator found that the AGD states that: **If the image verification fails, reboot the system after a few minutes. These few minutes will allow the image to be recovered from a redundant image. If the system does not boot up beyond this point, then the administrator is required to contact Evertz product support for further resolution.** **If fips self-test verification during boot failed following output is produced in console or syslog** **"Enabling fipscheck: Failed"** **The system allows you to boot beyond this point, but it is not operable in CC evaluated state. The administrator is required to contact Evertz product support for further assistance and resolution.**

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.

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

Verdict:
PASS.

5.1.5.4 FPT_TUD_EXT.1 TRUSTED UPDATE

5.1.5.4.1 FPT_TUD_EXT.1 TSS

The evaluator shall verify that the TSS describes 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.

Evaluator Findings:
The evaluator verified that the TSS describes how to query the currently active version. The TSS also describes how and when the inactive version becomes active. The evaluator verified this description. The relevant information is found in the following section(s): TOE Summary Specification **FPT_TUD_EXT.1** Upon investigation, the evaluator found that the TSS states that: **The site administrators do not have access to install any applications on the TOE. The EXE embedded system can only be updated with the valid firmware released by Evertz. The current firmware version is displayed on**

both webpage and in serial console menu. The TOE supports delayed activation of updates hence the inactive versions can be manually set to active by the Security Administrator on the web.

Once the desired image slot upgrade with the firmware binary is completed successfully, the administrator must manually change the "Next Boot Image" value from the current boot image to the newly installed image slot. On setting the next image, the firmware binary is extracted to the location that will be used during boot. Once extraction is complete, boot specific files are created.

The administrator must manually reboot for the new update to take effect.

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).

Evaluator Findings:

The evaluator verified that the TSS describes all TSF software update mechanisms for updating the system firmware and software.

The relevant information is found in the following section(s): TOE Summary Specification **FPT_TUD_EXT.1**

Upon investigation, the evaluator found that the TSS states that:
The first step of upgrading firmware image is to choose a desired image slot index. If any firmware image is pre-installed to the desired image slot, delete it. Continue using the image slot for firmware upload. After the firmware is uploaded, EXE will verify the firmware binary header with an Evertz-EXE-specific-file format header. If there is no mismatch, the new firmware code will be parsed for valid digital signatures.

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.

Evaluator Findings:

The evaluator verified that the TSS includes a description of the digital signature verification of the software before installation and that installation fails if the verification fails.

The relevant information is found in the following section(s): TOE Summary Specification **FPT_TUD_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

Verification of the firmware's digital signatures is performed using the public key stored on EXE. If unsuccessful, the firmware update file is rejected, and an error is displayed. The TSF does not provide an interface to change the local stored public key to administrators. If successful, firmware specific files are generated. Checksums of the firmware binary and firmware specific files are generated and stored under the

image slot chosen when uploading the firmware binary. The generated checksum is used to verify the firmware binary copy to the image slot location without compromising the integrity of the files.

If the digital signature fails, the upgrade fails, and a log event is generated. If the digital signature succeeds, the upgrade proceeds and the updated firmware is installed onto the TOE.

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.

Evaluator Findings:

The evaluator verified that the TSS describes the method by which the digital signature is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature of the update, and the actions that take place for both successful and unsuccessful signature verification.

The relevant information is found in the following section(s): TOE Summary Specification
FPT_TUD_EXT.1

Upon investigation, the evaluator found that the TSS states that:

The first step of upgrading firmware image is to choose a desired image slot index. If any firmware image is pre-installed to the desired image slot, delete it. Continue using the image slot for firmware upload. After the firmware is uploaded, EXE will verify the firmware binary header with an Evertz-EXE-specific-file format header. If there is no mismatch, the new firmware code will be parsed for valid digital signatures.

Once the desired image slot upgrade with the firmware binary is completed successfully, the administrator must manually change the "Next Boot Image" value from the current boot image to the newly installed image slot. On setting the next image, the firmware binary is extracted to the location that will be used during boot. Once extraction is complete, boot specific files are created.

Checksums for the extracted firmware files and boot specific files are created. The generated checksum is used to verify if the firmware files are extracted without compromising the integrity of the files.

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.

Evaluator Findings:

The options 'support automatic checking for updates' or 'support automatic updates' are not chosen from the selection in FPT_TUD_EXT.1.2, hence this requirement is not applicable.

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.

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

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.

Evaluator Findings:
The published hash is not used to protect the trusted update mechanism hence this activity is not applicable.

Verdict:
PASS.

5.1.5.4.2 FPT_TUD_EXT.1 AGD

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.

Evaluator Findings:
The evaluator verified that the guidance documentation describes how to query the currently active version. The guidance documentation also describes how to query the loaded but inactive version. The relevant information is found in the following section(s): **Verify Current Installed Image** and **Switch an Inactive Image to Active Image** Upon investigation, the evaluator found that the AGD states the **prerequisites and steps to verify the current image**. In addition, the **note on page 42** describes where inactive and active images are found. In addition, the section titled **'Switch an Inactive Image to Active Image'** in the AGD and verified that it **describes the necessary steps on how to activate the installed image in the next boot.**

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.

Evaluator Findings:
The evaluator verified that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification). The description includes the procedures for successful and unsuccessful verification. The description corresponds to the description in the TSS.

The relevant information is found in the following section(s): **Performing Secure Upgrade**

Upon investigation, the evaluator found that the AGD states that:

EXE supports secure upgrade to facilitate a robust and capable update of mechanisms in line with the standards set by the Common Criteria for Network Device Protection Profile. EXE supports the following features during any secure upgrade:

- **Multiple firmware version support simultaneously and simplified switch process between firmware versions.**
- **If the integrity or authenticity of the current image is faulted, the EXE will fail to boot.**
- **During the secure upgrade process, the integrity of the image is verified. If the verification fails, the failed image file is not created/mounted to the system and the image will not be available to be selected as the next boot image. The current boot image and the next boot image will remain to be the same current operational image.**
- **Image authenticity verification is done using digital Signature verification.**
- **Image Integrity validation is done using Signature verification and file corruption analysis.**

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.

Evaluator Findings:
Published hashes are not used. Hence, this is not applicable.

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.

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

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.

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

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.

Evaluator Findings:
Upon investigation, the evaluator examined the Security Target and verified that a certificate-based mechanism is not used for software update digital signature verification. Based on these findings, this assurance activity is considered not applicable.

Verdict:
PASS.

5.1.5.5 FPT_STM_EXT.1 RELIABLE TIME STAMPS

5.1.5.5.1 FPT_STM_EXT.1 TSS **[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.

Evaluator Findings:
The evaluator examined the TSS and ensured 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. The relevant information is found in the following section(s): TOE Summary Specification **FPT_STM_EXT.1** Upon investigation, the evaluator found that the TSS states that: **The TSF provides a reliable timestamp from the hardware clock on the TOE. Timestamps found in auditable log events use the system clock on EXE. In addition to the purpose of generating audit logs, this timestamp is used for the purposes of other time-sensitive operations on the TOE including cryptographic key regeneration intervals. Administrators can, as needed, set the system time clock through serial port console menu after each card reboot.** **Other functions which make use of timestamps include verification of X.509 certificate validity periods.** **The new system time is also used to set the hardware clock, which is a clock that runs independently of any control program running in the CPU and even when EXE is powered off. During EXE system startup, system time is initialized to the time from the hardware clock**.

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.

Evaluator Findings:
Upon investigation, the evaluator found that the “obtain time from the underlying virtualization system” is not selected in the ST, hence this is not applicable to the TOE.

Verdict:
PASS.

5.1.5.5.2 FPT_STM_EXT.1 AGD **[TD0632]**

The evaluator examines the guidance documentation to ensure it instructs the administrator how to set the time.

Evaluator Findings:
The evaluator examined the guidance documentation and ensured that it instructs the administrator how to set the time. The relevant information is found in the following section(s): **Configure System Date and Time** Upon investigation, the evaluator found that the AGD states the below steps: **Steps** 1. **Log in to the EXE serial console using “recovery” credentials.** 2. **Use the following to set the date of system.** **Once in the ‘Set Time’ section, time can be set by using the following format:** **YYYY-MM-DD hours:minutes:seconds** 3. **Press ENTER to apply the settings.**

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.

Evaluator Findings:
The TOE does not support the use of an NTP server; Hence, this activity is not applicable.

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.

Evaluator Findings:
Upon investigation, the evaluator found that the “obtain time from the underlying virtualization system” is not selected, Hence, this activity is not applicable.

Verdict:
PASS.

5.1.6 TOE ACCESS (FTA)

5.1.6.1 FTA_SSL_EXT.1 TSF-INITIATED SESSION LOCKING

5.1.6.1.1 FTA_SSL_EXT.1 TSS

The evaluator shall examine the TSS to determine whether local administrative session locking or termination is supported and the related inactivity time period settings.

Evaluator Findings:
The evaluator examined the TSS to determine whether local administrative session locking, or termination is supported and the related inactivity time period settings. The relevant information is found in the following section(s): TOE Summary Specification **FTA_SSL_EXT.1** Upon investigation, the evaluator found that the TSS states that: **Security Administrators can configure a maximum allowable period of inactivity for a Security Administrator session on the web interface or the local console. If there is no user interaction with the EXE for the specified amount of time, the session is terminated. The initial, default session timeout is 15 minutes. When the session is terminated, any unsaved changes will be discarded.**

Verdict:
PASS.

5.1.6.1.2 FTA_SSL_EXT.1 AGD

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.

Evaluator Findings:
The evaluator confirmed that the guidance documentation states whether local administrative session locking, or termination is supported and instructions for configuring the inactivity time period. Upon investigation, the evaluator found that the AGD states the required steps under **‘Set session Timeout’** section.

Verdict:
PASS.

5.1.6.2 FTA_SSL.3 TSF-INITIATED TERMINATION

5.1.6.2.1 FTA_SSL.3 TSS

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

Evaluator Findings:
The evaluator examined the TSS **FTA_SSL.3** and determined that it details the administrative remote session termination and the related inactivity time period. The relevant information is found in the following section(s): TOE Summary Specification **FTA_SSL.3** Upon investigation, the evaluator found that the TSS states that: **Security Administrators can configure a maximum allowable period of inactivity for a Security Administrator session on the web interface. The settings made on the web interface are applied to both local console and web interfaces. If there is no user interaction with the EXE for the specified amount of time, the session is terminated. The initial, default session timeout is 15 minutes. When the session is terminated, any unsaved changes will be discarded.**

Verdict:
PASS.

5.1.6.2.2 FTA_SSL.3 AGD

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

Evaluator Findings:
The evaluator confirmed that the guidance documentation includes instructions for configuring the inactivity time period for remote administrative session termination. Upon investigation, the evaluator found that the AGD states the required steps under **'Set session Timeout'** section.

Verdict:
PASS.

5.1.6.3 FTA_SSL.4 USER-INITIATED TERMINATION

5.1.6.3.1 FTA_SSL.4 TSS

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

Evaluator Findings:
The evaluator examined the TSS and determined that it details how the remote administrative session (and if applicable the local administrative session) are terminated.

The relevant information is found in the following section(s): TOE Summary Specification **FTA_SSL.4**

Upon investigation, the evaluator found that the TSS states that:

Administrators may terminate their own sessions by clicking "Logout" at the upper right hand of the web interface or by exiting the top-level menu on the console.

Verdict:
PASS.

5.1.6.3.2 FTA_SSL.4 AGD

The evaluator shall confirm that the guidance documentation states how to terminate a local or remote interactive session.

Evaluator Findings:

The evaluator confirmed that the guidance documentation states how to terminate a remote interactive session (and if applicable the local administrative session).

The relevant information is found in the following section(s): **Accessing the EXE**

Upon investigation, the evaluator found that the AGD states that:

Terminating Web Session

Remote:

Click "Logout" button on top right corner.

Terminating Serial Console Connection
Local:

Use the following until termination of the serial console connection.

#X

Verdict:
PASS.

5.1.6.4 FTA_TAB.1 DEFAULT TOE ACCESS BANNERS

5.1.6.4.1 FTA_TAB.1 TSS

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).

Evaluator Findings:

The evaluator checked the TSS and ensured that it details each administrative method of access (local and remote) available to the Security Administrator.

The relevant information is found in the following section(s): TOE Summary Specification **FIA_UIA_EXT.1/ FIA_UAU_EXT.2**

Upon investigation, the evaluator found that the TSS states that:

Administrators can log on via the web interface using HTTPS or locally on the serial port. A username and a password is required to authenticate the administrator for both methods. The Security Administrator is considered authenticated if the username and password match the stored credential values. For serial console, only the default 'recovery' user has access to the restricted shell.

Prior to successful identification and authentication, the TSF displays the TOE access banner specified in FTA_TAB.1. Users must acknowledge the warning banner before they can login to the system.

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).

Evaluator Findings:

The evidence about all administrative methods of access available to the Security Administrator is covered in the above activity.

The evaluator also verified that the TSS states that the TOE is displaying an advisory notice and a consent warning message for each administrative method of access.

The relevant information is found in the following section(s): TOE Summary Specification **FTA_TAB.1**

Upon investigation, the evaluator found that the TSS states that:

The TSF presents the access banner prior to authentication when a user connects to the remote web interface or local console CLI described in the FIA_UIA_EXT.1, FIA_UAU_EXT.2 description.

The TSF enables Security Administrators to alter the warning banner by navigating to the Perpetual User License Agreement tab on the web. From here the Security Administrator can modify the "Agree" text and/or the "Disagree" text. (The "Disagree" text shows up when a user "disagrees" with the Security Banner text. The banner can provide warnings against unauthorized access to the TOE as well as any other information that the Security Administrator wishes to communicate. Users who select "Disagree" are not permitted access to the TSF.

Verdict:
PASS.

5.1.6.4.2 FTA_TAB.1 AGD

The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message.

Evaluator Findings:
The evaluator examined the guidance documentation and ensured that it describes how to configure the banner message. The relevant information is found in the following section(s): **Configure Secure Access Banner** Upon investigation, the evaluator found that the AGD states the required steps in this section.

Verdict:
PASS.

5.1.7 TRUSTED PATH (FTP)

5.1.7.1 FTP_ITC.1 INTER-TSF TRUSTED CHANNEL

5.1.7.1.1 FTP_ITC.1 TSS

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.

Evaluator Findings:
The evaluator examined the TSS and determined 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 relevant information is found in the following section(s): TOE Summary Specification **FTP_ITC.1** Upon investigation, the evaluator found that the TSS states that: **The TSF communicates with the external syslog server using TLS as described in the descriptions of FAU_STG_EXT.1 and FCS_TLS* above. The TSF initiates the trusted channel with the Syslog server.** **The TSF communicates with a MAGNUM server (Video Switch Server) through TLS as well as described in the FCS_TLS* above. The MAGNUM server initiates the trusted channel with the TOE and is a trusted IT entity.**

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.

Evaluator Findings:
The evaluator also confirmed 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 relevant information is found in the following section(s): TOE Summary Specification **FTP_ITC.1**

Upon investigation, the evaluator found that the TSS have sufficient information to identify that TLS protocol is used for the two secure communication channels claimed.

Verdict:
PASS.

5.1.7.1.2 FTP_ITC.1 AGD

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.

Evaluator Findings:

The evaluator confirmed that the AGD sections **Configure TLS Server** and **Configure TLS Client** contain instructions for establishing the allowed protocols with each authorized IT entity.

In addition, the section titled '**Offloading Audit Logs'** section states that:

System log messages can be sent to a remote audit server. The remote audit server must listen on TCP Port 6514 for TLS connections, and its certificate chain must be trusted by EXE when the Secure Mode is enabled. All audit events are simultaneously sent to the remote server and the local store. If this or any outgoing client connection is unintentionally broken, EXE will automatically reconnect within seconds.

The section titled **'Configure TLS Server'** states that:

In case of an unexpected connection failure of the synergy server communication channel, the synergy server will wait for the connection from the TLS client (Evertz Magnum device, 3rd party video routers/source devices). If no data is received, the synergy server will reset the connection after the TCP session timeout limit is reached. For connection recovery instructions with Evertz Magnum, please refer to the Evertz Magnum CC guide. For recovery information of the channels with other 3rd party video source and destination streaming devices, please refer to the administrative guidance documents of those specific devices.

Verdict:
PASS.

5.1.7.2 FTP_TRP.1/ADMIN TRUSTED PATH

5.1.7.2.1 FTP_TRP.1/ADMIN TSS

The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected.

Evaluator Findings:
The evaluator examined the TSS and determined that the methods of remote TOE administration are indicated, along with how those communications are protected. The relevant information is found in the following section(s): TOE Summary Specification **FTP_TRP.1/Admin** **The TSF provides a trusted path for remote administration using HTTPS/TLS as described in FCS_HTTPS_EXT.1 and FCS_TLSS_EXT.1 descriptions. EXE uses encryption and restricts the choices of ciphers, hashes, and key-exchange algorithms to those allowed by the NDcPP.**

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.

Evaluator Findings:
The evaluator compared the protocols identified in the TSS to the definition of the SFR. The evaluator found that the protocols listed in the TSS are consistent with the protocols listed in the definition of the SFR.

Verdict:
PASS.

5.1.7.2.2 FTP_TRP.1/ADMIN AGD

The evaluator shall confirm that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method.

Evaluator Findings:
The evaluator confirmed that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method. The relevant information is found in the following section(s): **Login via Web GUI** Upon investigation, the evaluator found that the AGD contains instructions for establishing the remote HTTPS administrative sessions. It also describes the prerequisites.

Verdict:
PASS.

5.2 OPTIONAL REQUIREMENTS

5.2.1 CRYPTOGRAPHIC SUPPORT (FCS)

5.2.1.1 FCS_TLSS_EXT.2 EXTENDED: TLS SERVER SUPPORT FOR MUTUAL AUTHENTICATION

5.2.1.1.1 FCS_TLSS_EXT.2.1 AND FCS_TLSS_EXT.2.2 TSS

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

Evaluator Findings:
The ST -FIA_X509_EXT.2.1 claims that x509 certificates are used to support authentication for **TLS**. The evaluator ensured that the section **TOE Summary Specification** in the ST includes a description of the use of client-side certificates for TLS mutual authentication. Upon investigation, the evaluator found that the TSS states that: **For video switch control systems TLS trusted channels, the TOE requires TLS with mutual authentication.** **Instructions about generating/downloading CSR and loading certificate can be found in the EXE manual. The Administrator can only upload one certificate chain to include a single CA certificate. The same certificate will be used by EXE for both web service and MAGNUM control. The same CA will be used for certificate verification. EXE enforces mutual authentication and therefore requires client certificates to establish a connection.**

The evaluator shall verify the TSS describes how the TSF uses certificates to authenticate the TLS client.

Evaluator Findings:
The evaluator verified the section **TOE Summary Specification** in the ST describes how the TSF uses certificates to authenticate the TLS client. Upon investigation, the evaluator found that the TSS states that: **When validating a server's certificate or the client's certificate in mutual authentication, EXE uses CRL (certification revocation list) to check for invalid certificates. The TOE pulls the CRL file from the CRL-DP to use by EXE during certificate validation process to check for revocation status of the peer certificates.** **EXE allows configuration of an RFC 6125 reference identifier from a peer it expects to connect with before connection is made. The reference identifier is matched to either the CN or the SAN in the certificate presented for authentication. The verification against peer certificate is implemented within OpenSSL using a bitwise comparison of the DN and SAN-DNS field. IP addresses are not supported as reference identifiers. EXE supports FQDN identifier types only. SRV-ID and URI-ID types are not supported.** **For all the TLS client and server connections, with the exception of 'revocation status verification failures', if the certificate verification fails for any other reason (including a failure to establish a connection), the connection attempt fails, and the trusted channel is not established. There are no fallback authentication functions for failed certificate authentication.**

The evaluator shall verify the TSS describes if the TSF supports any fallback authentication functions (e.g. username/password, challenge response) the TSF uses to authenticate TLS clients that do not present a

certificate. If fallback authentication functions are supported, the evaluator shall verify the TSS describes whether the fallback authentication functions can be disabled.

Evaluator Findings:
The evaluator reviewed the section **TOE Summary Specification** in the ST and found that the TSS states that: **There are no fallback authentication functions for failed certificate authentication.** The certificate authentication mechanism is described in FIA_X509_EXT.1, FIA_X509_EXT.2, and FIA_X509_EXT.3 entries on the TSS.

Verdict:
PASS.

5.2.1.1.2 FCS_TLSS_EXT.2.3 TSS

The evaluator shall verify that the TSS describes which types of identifiers are supported during client authentication (e.g. Fully Qualified Domain Name (FQDN)). If FQDNs are supported, the evaluator shall verify that the TSS describes that corresponding identifiers are matched according to RFC6125. For all other types of identifiers, the evaluator shall verify that the TSS describes how these identifiers are parsed from the certificate, what the expected identifiers are and how the parsed identifiers from the certificate are matched against the expected identifiers.

Evaluator Findings:
The evaluator examined the section titled **TOE Summary Specification** in the Security Target. Upon investigation, the evaluator found that the TSS states that: **EXE allows configuration of an RFC 6125 reference identifier from a peer it expects to connect with before connection is made. The reference identifier is matched to either the CN or the SAN in the certificate presented for authentication. The verification against peer certificate is implemented within OpenSSL using a bitwise comparison of the DN and SAN-DNS field. IP addresses are not supported as reference identifiers.**

Verdict:
PASS.

5.2.1.1.3 FCS_TLSS_EXT.2.1 AND FCS_TLSS_EXT.2.2 AGD

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

Evaluator Findings:
The evaluator examined the section titled **Configure TLS Server** in the AGD to verify that it describes the certificate configuring instructions for TLS mutual authentication. The note on page 30 states the following: **"Reference identifier is only used for synergy server communication with mutual authentication. No additional configuration is required for mutual authentication. The EXE will use mutual authentication for connection requests that are received from the configured reference identifier."**

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

Evaluator Findings:
The evaluator examined the section titled **Configure TLS Server** in the AGD and found that the AGD states that: **For all the TLS client and server connections, with the exception of 'revocation status verification failures', if the certificate verification fails for any other reason (including a failure to establish a connection), the connection attempt fails, and the trusted channel is not established. There are no fallback authentication functions for failed certificate authentication.**

Verdict:
PASS.

5.2.1.1.4 FCS_TLSS_EXT.2.3 AGD

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

Evaluator Findings:
The evaluator examined the section titled "**Configure TLS Server"** in the AGD to verify that it contains any configuration necessary to meet the requirement. Upon investigation, the evaluator found that the AGD states that: **Only host names are used for reference identifiers, the product does not support IPV4 and IPv6 addressing in reference identifier. EXE allows configuration of reference identifier from a peer it expects to connect with before connection is made. The reference identifier can be any string up to 64 bytes that is present in the peer certificate's CN/SAN field. The verification against CN/SAN peer certificate is implemented within OpenSSL. A wildcard in the left-most label in the certificate will allow a successful connection, but a reference identifier without a left-most label as in the certificate, the connection will fail, i.e., awesome.com doesn't match *.awesome.com.** **Reference identifier is only used for synergy server communication with mutual authentication. No additional configuration is required for mutual authentication. The EXE will use mutual authentication for connection requests that are received from the configured reference identifier.**

Verdict:
PASS.

5.3 SELECTION-BASED REQUIREMENTS

5.3.1 CRYPTOGRAPHIC SUPPORT (FCS)

5.3.1.1 FCS_HTTPS_EXT.1 HTTPS PROTOCOL

5.3.1.1.1 FCS_HTTPS_EXT.1 TSS

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

Evaluator Findings:
The evaluator examined the section **TOE Summary Specification** in the ST and determine that enough detail is provided to explain how the implementation complies with RFC 2818. Upon investigation, the evaluator found that the TSS states that: **The TOE acts as a TLS/HTTPS server to provide web access to administrators. The TOE's HTTPS functionality is in accordance with all should statements in RFC 2818.** **The TSF only supports TLSv1.2 for HTTPS/TLS. Connection requests that include SSL 2.0, SSL 3.0, TLS 1.0 or TLS 1.1 are denied. If the TSF receives a ClientHello message that requests TLSv1.1 or earlier, the TSF sends a fatal handshake failure message and terminates the connection**.

Verdict:
PASS.

5.3.1.1.2 FCS_HTTPS_EXT.1 AGD

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.

Evaluator Findings:
The evaluator examined the guidance documentation, section **Configure TLS Server** and verified that it instructs the Administrator how to configure TOE for use as an HTTPS server.

Verdict:
PASS.

5.3.1.2 FCS_TLSC_EXT.1 EXTENDED: TLS CLIENT PROTOCOL WITHOUT MUTUAL AUTHENTICATION

5.3.1.2.1 FCS_TLSC_EXT.1.1 TSS

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

Evaluator Findings:
The evaluator checked the description of the implementation of this protocol in the section **TOE Summary Specification** in the ST and ensured that the ciphersuites supported are specified. Upon investigation, the evaluator found that the TSS states that: **EXE specifies only a restricted set of cipher suites that it supports during the negotiation phase with a client or a server. If no match of cipher suites can be found with peer, TLS session will not be started. The following cipher suites are supported:** • **TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268** • **TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268** • **TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246** • **TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246** • **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289** • **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289** **EXE supports cipher suites that use ECDHE and RSA schemes for key exchange and RSA keys for authentication.**

The evaluator shall check the TSS to ensure that the ciphersuites specified include those listed for this component.

Evaluator Findings:
The evaluator ensured that the ciphersuites claimed in the ST section 5.2.2.10 matches the ciphersuites listed in the section **TOE Summary Specification** in the ST. The evaluator also ensured that the ciphersuites claimed in the ST are present in section B.3.1.6 of NDcPP.

Verdict:
PASS.

5.3.1.2.2 FCS_TLSC_EXT.1.2 TSS

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.

Evaluator Findings:
The evaluator ensured that the section **TOE Summary Specification** in the ST 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 and whether IP addresses and wildcards are supported. Upon investigation, the evaluator found that the TSS states that: **The reference identifier is matched to either the CN or the SAN in the certificate presented for authentication. The verification against peer certificate is implemented within OpenSSL using a bitwise comparison of the DN**

and SAN-DNS field. IP addresses are not supported as reference identifiers. EXE supports FQDN identifier types only. SRV-ID and URI-ID types are not supported.

EXE does not support certificate pinning.

EXE supports wildcard in certificates. The wildcard must be in the left-most label of the presented identifier and can only cover one level of subdomains. For the reference identifier without a left-most label as in the certificate, the connection will fail, i.e., awesome.com doesn't match *.awesome.com.

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.

Evaluator Findings:
Not applicable because the TOE is not a distributed TOE.

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.

Evaluator Findings:
Upon investigation, the evaluator found that the section **TOE Summary Specification** in the ST states that: **IP addresses are not supported as reference identifiers.** Hence, this requirement is not applicable.

The evaluator shall also ensure that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced.

Evaluator Findings:
Upon investigation, the evaluator found that the section **TOE Summary Specification** in the ST states that: **IP addresses are not supported as reference identifiers.** Hence, this requirement is not applicable.

Verdict:
PASS.

5.3.1.2.3 FCS_TLSC_EXT.1.4 TSS

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.

Evaluator Findings:
The evaluator verified that section **TOE Summary Specification** in the ST describes the Supported Elliptic Curves/Supported Groups Extension and whether the required behaviour is performed by default or may be configured. Upon investigation, the evaluator found that the TSS states that: **The elliptic curve Diffie Hellman and RSA are supported for key establishment in TLS for both client and server. The RSA key establishment uses keys with key-sizes 2048 bits, 3072 bits, and 4096 bits. EC-DH key establishment uses NIST curves, P-256, P-384, and P-521. By default, the TOE presents the supported Elliptic Curve Extensions, secp256r1, secp384r1, and secp521r1 in the Client Hello. The TOE conforms to RFC 5246, section 7.4.3 for key exchange.**

Verdict:
PASS.

5.3.1.2.4 FCS_TLSC_EXT.1.1 AGD

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.

Evaluator Findings:
The evaluator checked the AGD section **Configure TLS Client** and ensured that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

Verdict:
PASS.

5.3.1.2.5 FCS_TLSC_EXT.1.2 AGD

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

Evaluator Findings:
The evaluator ensured that the AGD section **Configure TLS Client** 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). Upon investigation, the evaluator found that the AGD states that: **Only host names are used for reference identifiers, EXE does not support IPv4 or IPv6 addressing in reference identifier. EXE allows configuration of reference identifier from a peer it expects to connect with before**

connection is made. The reference identifier can be any string up to 64 bytes that is present in the peer certificate's CN/SAN field. The verification against CN/SAN peer certificate is implemented within OpenSSL. A wildcard in the left-most label in the certificate will allow a successful connection, but a reference identifier without a left-most label as in the certificate, the connection will fail, i.e., awesome.com doesn't match *.awesome.com.

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 AGD provides instructions for establishing unique reference identifiers based on RFC5280 attributes.

Evaluator Findings:
The TOE is not a distributed TOE; hence this activity is not applicable.

Verdict:
PASS.

5.3.1.2.6 FCS_TLSC_EXT.1.4 AGD

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

Evaluator Findings:
The evaluator verified that the AGD sections **Key Parameters** and **Cipher Suites** include information about the configuration of the Supported Elliptic Curves/Supported Groups Extension. Upon investigation, the evaluator found that the AGD states that: **EXE does not allow or provide interfaces for the administrator to configure/enable/disable cipher suites. Rather EXE by default supports the following cipher suites in compliance with CC evaluation criteria implicitly. No configuration is needed or possible in both cipher suites selection and RNG.** **EXE accepts 2048-bits, 3072-bits, and 4096-bits RSA keys from the TLS Clients and TLS Servers (with mutual authentication) but EXE only generates 2048-bit RSA keys during Certificate Signing Request generation. EXE does not allow or provide interfaces for the administrator to configure key parameters such as the RSA key size or elliptic curves. Parameters are hard coded implicitly in accordance with the CC evaluation criteria.**

Verdict:
PASS.

5.3.1.3 FCS_TLSS_EXT.1 EXTENDED: TLS SERVER PROTOCOL WITHOUT MUTUAL AUTHENTICATION

5.3.1.3.1 FCS_TLSS_EXT.1.1 TSS

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

Evaluator Findings:
The evaluator checked the description of the implementation of this protocol in the section **TOE Summary Specification** in the ST and ensured that the ciphersuites supported are specified. Upon investigation, the evaluator found that the TSS states that: **EXE specifies only a restricted set of cipher suites that it supports during the negotiation phase with a client or a server. If no match of cipher suites can be found with peer, TLS session will not be started. The following cipher suites are supported:** • **TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268** • **TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268** • **TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246** • **TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246** • **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289** • **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289** **EXE supports cipher suites that use ECDHE and RSA schemes for key exchange and RSA keys for authentication.**

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

Evaluator Findings:
The evaluator ensured that the ciphersuites claimed in the ST section 5.2.2.10 are identical to the ciphersuites listed in the section **TOE Summary Specification** in the ST. The evaluator also ensured that the ciphersuites claimed in the ST are present in section B.3.1.6 of NDcPP.

Verdict:
PASS.

5.3.1.3.2 FCS_TLSS_EXT.1.2 TSS

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

Evaluator Findings:
The evaluator verified that the section **TOE Summary Specification** in the ST contains a description of how the TOE technically prevents the use of old SSL and TLS versions. Upon investigation, the evaluator found that the TSS states that: **The TSF only supports TLSv1.2 for HTTPS/TLS. Connection requests that include SSL 2.0, SSL 3.0, TLS 1.0 or TLS 1.1 are denied. If the TSF receives a ClientHello message that requests TLSv1.1 or earlier, the TSF sends a fatal handshake failure message and terminates the connection.**

Verdict:
PASS.

5.3.1.3.3 FCS_TLSS_EXT.1.3 TSS **[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.

Evaluator Findings:
The evaluator verified that the section **TOE Summary Specification** in the ST 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. Upon investigation, the evaluator found that the TSS states that: **The elliptic curve Diffie Hellman and RSA are supported for key establishment in TLS for both client and server.** **EC-DH key establishment uses NIST curves, P-256, P-384, and P-521. By default, the TOE presents the supported Elliptic Curve Extensions, secp256r1, secp384r1, and secp521r1 in the Client Hello. The TOE conforms to RFC 5246, section 7.4.3 for key exchange.** **The following cipher suites are supported:** • **TLS_RSA_WITH_AES_128_CBC_SHA** • **TLS_RSA_WITH_AES_256_CBC_SHA** • **TLS_RSA_WITH_AES_128_CBC_SHA256** • **TLS_RSA_WITH_AES_256_CBC_SHA256** • **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256** • **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384**

Verdict:
PASS.

5.3.1.3.4 FCS_TLSS_EXT.1.4 TSS **[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).

Evaluator Findings:
The evaluator verified that the section **TOE Summary Specification** in the ST and found that session resumption (based on session IDs or session tickets) is not supported. Upon investigation, the evaluator found that the TSS states that: **EXE does not support session resumption based on session IDs or session tickets.**

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.

Evaluator Findings:
The evaluator verified that the section **TOE Summary Specification** in the ST and found that session resumption (based on session IDs or session tickets) is not supported. Hence, this is not applicable to the TOE.

The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.

Evaluator Findings:
The evaluator verified that the section **TOE Summary Specification** in the ST and found that session resumption (based on session IDs or session tickets) is not supported. Hence, this is not applicable to the TOE.

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.

Evaluator Findings:
The evaluator verified that the section **TOE Summary Specification** in the ST and found that session resumption (based on session IDs or session tickets) is not supported. Hence, this is not applicable to the TOE.

If the TOE claims a (D)TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator shall verify 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.

Evaluator Findings:
The evaluator verified that the section **TOE Summary Specification** in the ST and found that session resumption (based on session IDs or session tickets) is not supported. Hence, this is not applicable to the TOE.

Verdict:
PASS.

5.3.1.3.5 FCS_TLSS_EXT.1.1 AGD

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).

Evaluator Findings:
The evaluator checked the AGD section **Configure TLS Server** and ensured that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

Verdict:
PASS.

5.3.1.3.6 FCS_TLSS_EXT.1.2 AGD

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

Evaluator Findings:
The evaluator verified that any configuration necessary to meet the requirement must be contained in the AGD section **Configure TLS Server**. The relevant information is found in the following section(s): **Configure TLS Server** Upon investigation, the evaluator found that the AGD states that: **In EXE, both WebGUI and Synergy Server (Magnum) use TLS Server capabilities to provide secure communication between the clients and server. The TLS Server comes with the following functionalities:** - **Supports ONLY TLSv1.2** - **SSLv3 and SSLv2 ARE NOT supported.** - **Implicit cipher suite selection** - **Implicit Key-Exchange selection**

Verdict:
PASS.

5.3.1.3.7 FCS_TLSS_EXT.1.3 AGD

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

Evaluator Findings:
The evaluator verified that any configuration necessary to meet the requirement must be contained in the AGD sections titled "**Cipher Suites", "Key Parameters", "Hash and Keyed-Hash Algorithms"**. Upon investigation, the evaluator found that on all these sections it is stated that these parameters are non-configurable and supported by default.

Verdict:
PASS.

5.3.1.3.8 FCS_TLSS_EXT.1.4 AGD **[TD0569]**

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

Evaluator Findings:
According to the ST, session resumption is not supported by the TOE by default. The evaluator reviewed the AGD and did not find any evidence to disable session resumption.

Verdict:
PASS.

5.3.2 IDENTIFICATION AND AUTHENTICATION (FIA)

5.3.2.1 FIA_X509_EXT.1/REV X.509 CERTIFICATE VALIDATION

5.3.2.1.1 FIA_X509_EXT.1/REV TSS

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).

Evaluator Findings:
The evaluator ensured 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. The relevant information is found in the following section(s): TOE Summary Specification **FIA_X509_EXT.1/Rev** Upon investigation, the evaluator found that the TSS states that: **EXE uses OpenSSL for X.509 certificate validation. The certificate path is validated by ensuring that all the CA certificates have the basicConstraints extension and the path must terminate with a trusted CA certificate. The extendedKeyUsage on each certificate is also checked to ensure there is no inappropriate usage. Server certificates must have the Server Authentication purpose, client's certificates must have the Client Authentication purpose. Certificates for code signing and OCSP signing are not used or accepted by the TOE. Each certificate (other than the first certificate) in the certificate chain has the Subject Type=CA flag set. Certificates are not used for any purposes other than establishing TLS sessions.** **If certificates are uploaded to EXE for its own use those certificates are checked upon upload. When the TOE acts as a server, it does not perform verification of its server certificate. The TOE's client certificate is validated prior to use for authentication as well as upon upload. The certificate presented by remote TLS**

clients using mutual authentication is validated during the establishment of a TLS connection. The full certificate chain presented by TLS servers are validated during the establishment of a TLS connection.

For an expired certificate, EXE will deny the connection. EXE also uses CRL to verify whether the leaf certificate or intermediate CA certificate have been revoked. During session establishment with EXE, any byte modification in the certificate will lead to the failure of connection.

The TSF verifies the validity of a certificate when:

- **A TLS client establishes a TLS connection with mutual authentication.**
- **A TLS server presents certificates to the TOE as a part of a TLS connection.**

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.

Evaluator Findings:

The TSS describes when revocation checking is performed and on what certificates. The relevant information is found in the following section(s): TOE Summary Specification **FIA_X509_EXT.1/Rev**

Upon investigation, the evaluator found that the TSS states that:

EXE also uses CRL to verify whether the leaf certificate or intermediate CA certificate have been revoked. During session establishment with EXE, any byte modification in the certificate will lead to the failure of connection.

The TSF verifies the validity of a certificate when:

- **A TLS client establishes a TLS connection with mutual authentication.**
- **A TLS server presenting certificates to the TOE as a part of a TLS connection.**

Verdict:
PASS.

5.3.2.1.2 FIA_X509_EXT.1/REV AGD

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.

Evaluator Findings:

The evaluator also ensured that the AGD describes where the check of validity of the certificates takes place and describes how certificate revocation checking is performed and on which certificate.

The relevant information is found in the following section(s): **Certificate Management**

Upon investigation, the evaluator found that the AGD states that:

- **The certificate path is validated by ensuring that all the CA certificates have the basicConstraints extension, and the path must terminate with a trusted CA certificate.**
- **The extended Key Usage on each certificate is checked to ensure there is no inappropriate usage.**
- **Server certificates must have the Server Authentication purpose, client's certificates must have the Client Authentication purpose.**
- **Certificates for code signing and OCSP signing are not used or accepted by the TOE. Each certificate (other than the first certificate) in the certificate chain has the Subject Type=CA flag set.**
- **If certificates are uploaded to EXE for its own use those certificates are checked upon upload. When the TOE acts as a server it does not perform verification of its own server certificate. The TOE's client certificate is validated prior to use for authentication as well as upon upload. The certificate presented by remote TLS clients using mutual authentication is validated during the establishment of a TLS connection.**
- **For an expired certificate, EXE will deny the connection.**
- **EXE also uses CRL to verify whether the leaf certificate or intermediate CA certificate has been revoked. During session establishment with EXE, any byte modification in the certificate will lead to the failure of connection.**

Verdict:
PASS.

5.3.2.2 FIA_X509_EXT.2 X.509 CERTIFICATE AUTHENTICATION

5.3.2.2.1 FIA_X509_EXT.2 TSS

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.

Evaluator Findings:

The evaluator checked the TSS and ensured that it describes how the TOE chooses which certificates to use, and any necessary instructions in the AGD for configuring the operating environment so that the TOE can use the certificates.

The relevant information is found in the following section(s): TOE Summary Specification
FIA_X509_EXT.2

Upon investigation, the evaluator found that the TSS states that:
Instructions about generating/downloading CSR and loading certificate can be found in [EXE CC Admin Guide]. The Administrator can only upload one certificate chain to include a single CA certificate. The same certificate will be used by EXE for both web service and MAGNUM control. The same CA will be used for certificate verification. EXE enforces mutual authentication and therefore requires client certificates to establish a connection.

The evaluator shall examine the TSS and 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.

Evaluator Findings:
The evaluator examined the TSS and confirmed 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 relevant information is found in the following section(s): TOE Summary Specification **FIA_X509_EXT.2** Upon investigation, the evaluator found that the TSS states that: **The CRLs are obtained from a CRL distribution point over HTTP and are refreshed according to the default CRL update-interval. If the TOE is unable to reach the CRL DP it will accept the certificate and the session associated with the certificate will be established.**

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 AGD contains instructions on how this configuration action is performed.

Evaluator Findings:
The evaluator verified that any distinctions between trusted channels are described in the TSS. The relevant information is found in the following section(s): TOE Summary Specification **FIA_X509_EXT.2** Upon investigation, the evaluator found that the TSS states that: **The Administrator can only upload one certificate chain to include a single CA certificate. The same certificate will be used by EXE for both web service and MAGNUM control. The same CA will be used for certificate verification. EXE enforces mutual authentication and therefore requires client certificates to establish a connection.** The evaluator verified that the AGD does not mention any configuration of the default action being configurable. AGD sections **"Configure TLS Server"** and **"Configure TLS Client"** state that by default, the connection will be 'accepted' if the revocation status verification fails.

Verdict:
PASS.

5.3.2.2.2 FIA_X509_EXT.2 AGD

The evaluator shall also ensure that the guidance documentation describes the configuration required in the operating environment so the TOE can use the certificates.

Evaluator Findings:
The evaluator also ensured that the AGD describes the configuration required in the operating environment so the TOE can use the certificates. The relevant information is found in the following section(s): **Configure TLS Server** and **Configure TLS Client** Upon investigation, the evaluator found that the AGD describes all the prerequisites and configuration steps that are required for the EXE to use the certificates for each TLS connection. This section also states that: **For all the TLS client and server connections, if the certificate verification fails for any reason (including a failure to establish a connection), the connection attempt fails, and the trusted channel is not established. There are no fallback authentication functions for failed certificate authentication. The administrators must refer to the audit logs to identify what causes the failure. The detailed audit log description can be found in the 'Audit Events' section below.**

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.

Evaluator Findings:
The evaluator also ensured that the AGD describes the configuration required in the operating environment so the TOE can use the certificates. The AGD also includes any required configuration on the TOE to use the certificates. The relevant information is found in the following section(s): **Configure TLS Server** and **Configure TLS Client** Upon investigation, the evaluator found that the AGD also describes the behaviour when the connection cannot be established during the validity check of a certificate used in establishing a trusted channel. Both these sections state that: **For all the TLS client and server connections, with the exception of 'revocation status verification failures', if the certificate verification fails for any other reason (including a failure to establish a connection), the connection attempt fails, and the trusted channel is not established. There are no fallback authentication functions for failed certificate authentication. If the EXE is unable to reach a CRL Distribution Point, it will accept the certificate and the session associated with the certificate will be established, however, a log is generated indicating the reason for validation failure. The administrators must refer to the audit logs to identify what caused the failure.**

Verdict:
PASS.

5.3.2.3 FIA_X509_EXT.3 EXTENDED: X509 CERTIFICATE REQUESTS

5.3.2.3.1 FIA_X509_EXT.3 TSS

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.

Evaluator Findings:
The ST does not claim “device-specific information”. Hence, this assurance activity is considered not applicable to the TOE.

Verdict:
PASS.

5.3.2.3.2 FIA_X509_EXT.3 AGD

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.

Evaluator Findings:
The evaluator checked and ensured that the AGD contains instructions on requesting certificates from a CA, including generation of a Certificate Request. The relevant information is found in the following section(s): **Configure TLS Server** Upon investigation, the evaluator found that the AGD states that: **EXE does not allow the configuration of CSR parameters; the following default parameters are used. These parameters will be customizable starting v1.7.-** • **Country Name: Canada** • **State or Province Name: Ontario** • **Locality Name: Burlington** • **Organization Name: Evertz Microsystems Ltd.** • **Organizational Unit Name: EXE** • **Common Name: Configured primary IP address of EXE** • **Email Address: support@evertz.com**

Verdict:
PASS.

5.3.3 SECURITY MANAGEMENT (FMT)

5.3.3.1 FMT_MOF.1/FUNCTIONS MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR

5.3.3.1.1 FMT_MOF.1/FUNCTIONS TSS

For distributed TOEs see Section 2.4.1.1.

Evaluator Findings:
The TOE is not distributed; Hence, this requirement is not applicable.

For non-distributed TOEs, the evaluator shall ensure the TSS for each administrative function identified the TSS details how the Security Administrator determines or modifies the behaviour of (whichever is supported by the TOE) transmitting audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full (whichever is supported by the TOE).

Evaluator Findings:
The evaluator examined the TSS and ensured that it details how the Security Administrator determines or modifies the behaviour of transmitting audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full. The relevant information is found in the following section(s): **FMT_MOF.1/Functions** Upon investigation, the evaluator found that the TSS states that: **EXE gives the Security Administrator the ability to manage the security functions: auditing operations, ….** **Information on how a Security Administrator can manage Audit Operations is described in this table under FAU_STG_EXT.1 above.** The evaluator examined section FAU_STG_EXT.1 as described and found following information: **Information is also sent (using TLS 1.2) to an external Syslog server. For this to happen, an external syslog server should be configured (IP address/TCP Port number). A trusted certificate chain that is used to sign syslog server's certificate must also be uploaded to EXE.**

Verdict:
PASS.

5.3.3.1.2 FMT_MOF.1/FUNCTIONS AGD

For distributed TOEs see Section 2.4.1.2.

Evaluator Findings:
The TOE is not distributed; Hence, this requirement is not applicable.

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the Security Administrator determines or modifies the behaviour of (whichever is supported by the TOE) transmitting audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full (whichever is supported by the TOE) are performed to include required configuration settings.

Evaluator Findings:
The evaluator examined the AGD and ensured that it describes how the Security Administrator determines or modifies the behaviour of transmitting audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full are performed to include required configuration settings. The section "**Audit Events"** in the AGD states that no configuration is required by administrators for local audit event generation. It also explains that the default behaviour when the local audit storage space is full cannot be modified by the administrators. Additionally, this section states that the audit storage path cannot be accessed by the recovery user through the console. The configuration steps on how to modify the behaviour of transmitting audit data to an external IT entity is described in the section - **Offloading Audit Logs**.

Verdict:
PASS.

5.3.3.2 FMT_MTD.1/CRYPTOKEYS MANAGEMENT OF TSF DATA

5.3.3.2.1 FMT_MTD.1/CRYPTOKEYS TSS

For distributed TOEs see Section 2.4.1.1.

Evaluator Findings:
The TOE is not distributed; Hence, this requirement is not applicable.

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.

Evaluator Findings:
The evaluator examined the TSS and ensured that it lists the keys the Security Administrator is able to manage to include the options available and how those operations are performed. The relevant information is found in the following section(s): **FMT_MTD.1/CryptoKeys** Upon investigation, the evaluator found that the TSS states that: **The CLI allow the Security Administrator to perform the following TSF management functions on cryptographic keys:** • **TLS Key Generation (TLS keys are automatically generated when creating a CSR)** • **TLS Key Reset/Replacement (When a CSR is generated, previous TLS key will be deleted and replaced by the new key. The TLS keys cannot be imported from outside the TOE. The administrators cannot delete TLS keys manually).**

Verdict:
PASS.

5.3.3.2.2 FMT_MTD.1/CRYPTOKEYS AGD

For distributed TOEs see Section 2.4.1.2.

Evaluator Findings:
The TOE is not distributed; Hence, this requirement is not applicable.

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.

Evaluator Findings:
The evaluator examined the AGD and ensured that it lists the keys the Security Administrator is able to manage to include the options available and how those operations are performed. The relevant information is found in the following section(s): TOE Summary Specification **Key Parameters** Upon investigation, the evaluator found that the AGD states that: **EXE accepts 2048-bits, 3072-bits, and 4096-bits RSA keys from the TLS Clients and TLS Servers (with mutual authentication) but EXE only generates 2048-bit RSA keys during Certificate Signing Request generation. EXE does not allow or provide interfaces for the administrator to configure key parameters such as the RSA key size; Parameters are hard coded implicitly in accordance with the CC evaluation criteria.**

Verdict:
PASS.

6 SECURITY ASSURANCE REQUIREMENTS

6.1 ADV: DEVELOPMENT

6.1.1 BASIC FUNCTIONAL SPECIFICATION (ADV_FSP.1)

6.1.1.1 (5.2.1.1) EVALUATION ACTIVITY

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.

In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g. audit review or performing updates). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. The intent is that these interfaces will be adequately tested and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied.

The set of TSFI that are provided as evaluation evidence are contained in the Administrative Guidance and User Guidance.

Evaluator Findings:
TOE Design information that can be made public is available in the guidance documentation and in the ST. Any sensitive or proprietary information required by this protection profile is not to be made public. It is not necessary to provide a complete specification of the TSFIs. For NDcPP, additional “functional specification” documentation is not necessary because this requirement is satisfied by multiple other documents (AGD, TSS, and Testing). All associated activities are covered in the Test Report, ST, and AGD documents. NDcPP2.2e, section 7.2.1 states that: “For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation.” All of the above information is applicable to the ADV Evaluation Activities (5.2.1.1, 5.2.1.2, and 5.2.1.3) in NDcPP2.2e-SD. The evaluator examined the ST (Security Target) and the AGD (interface documentation) to verify that it describes the purpose and method of use for each TSFI that is identified as being security relevant. The evaluator examined the entire AGD. The evaluator verified the AGD describes the purpose and method of use for each security relevant TSFI by verifying the AGD satisfies all the AGD Evaluation Activities. During testing, the evaluator used the product and its interfaces extensively and did not find any areas that were deficient.

Verdict:
PASS.

6.1.1.2 (5.2.1.2) EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator checked the interface documentation (AGD) and ensured it identifies and describes the parameters for each TSFI that is identified as being security relevant. This is covered in the previous evaluation activity above.

Verdict:
PASS.

6.1.1.3 (5.2.1.3) EVALUATION ACTIVITY

The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

The evaluator uses the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2, including the EAs associated with testing of the interfaces.

It should be noted that there may be some SFRs that do not have an interface that is explicitly “mapped” to invoke the desired functionality. For example, generating a random bit string, destroying a cryptographic key that is no longer needed, or the TSF failing to a secure state, are capabilities that may be specified in SFRs, but are not invoked by an interface.

However, if the evaluator is unable to perform some other required EA because there is insufficient design and interface information, then the evaluator is entitled to conclude that an adequate functional specification has not been provided, and hence that the verdict for the ADV_FSP.1 assurance component is a ‘fail’.

Evaluator Findings:
The evaluator examined the interface documentation to develop a mapping of the interfaces to SFRs. The evaluator used the provided documentation to first identify, and then examine a representative set of interfaces to perform the evaluator activities presented in Section 2, including the evaluation activities associated with testing of the interfaces. This is covered in the previous evaluation activity above.

Verdict:
PASS.

6.2 AGD: GUIDANCE DOCUMENTS

6.2.1 OPERATIONAL USER GUIDANCE (AGD_OPE.1)

6.2.1.1 (5.3.1.1) EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator checked the requirements above are met by the AGD. The AGD is distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration. Upon investigation, the evaluator found that the CC guidance will be published with the CC certificate on www.niap-ccevs.org.

Verdict:
PASS.

6.2.1.2 (5.3.1.2) EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator ensured that the AGD is provided for every Operational Environment that the product supports as claimed in the Security Target. The section titled of the AGD was used to determine the verdict of this assurance activity. The AGD specifies that the platforms supported are: • **NATX-8-100G-CC** • **NATX-16-100G-CC** • **NATX-32-100G-1-CC** • **NATX-64-100G-2-CC** • **MMA10G-NATX-8-CC** • **MMA10G-NATX-16-CC** • **MMA10G-NATX-32-CC** • **MMA10G-NATX-64-CC** • **MMA10G-IPX128** • **3080IPX-48-25G-CC** The AGD covers all the models claimed in the ST and some additional models as well.

Verdict:
PASS.

6.2.1.3 (5.3.1.3) EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator ensured that the AGD contains instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It provides a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.

Verdict:
PASS.

6.2.1.4 (5.3.1.4) EVALUATION ACTIVITY

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.

Evaluator Findings:
The entire AGD was used to determine the verdict of this work unit. Each confirmation command indicates tested options. Additionally, covers configuration of the in-scope functionality where additional configuration might be required. The evaluator ensured the AGD makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

Verdict:
PASS.

6.2.1.5 (5.3.1.5) EVALUATION ACTIVITY **[TD0536]**

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

- The AGD 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.

- **[TD0536]** 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:
    - Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).
    - Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes instructions that describe at least one method of validating the hash/digital signature.

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

Evaluator Findings:
The evaluator verified the guidance documentation contains instructions for configuring any cryptographic engines in AGD_OPE.1 Test #3. The evaluator verified the guidance documentation describes the process for verifying updates in FPT_TUD_EXT.1 Guidance 2. The evaluator verified the guidance documentation makes it clear which security functionality is covered by the Evaluation Activities in AGD_OPE.1 Test #4.

Verdict:
PASS.

6.2.2 PREPARATIVE PROCEDURES (AGD_PRE.1)

6.2.2.1 (5.3.2.1) EVALUATION ACTIVITY

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).

The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE product itself).

Evaluator Findings:
The evaluator examined the Preparative procedures to ensure they include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality. The evaluator reviewed the sections titled **'Operational Environment' and 'Obtaining and Installing the CC Certified Firmware'** of the AGD. The evaluator found that these sections describe how the Operational Environment must meet: **OE.PHYSICAL** is covered by an explicit statement in the CC Guide. Note that the evaluator believes, generally, speaking, that OE.NO_GENERAL_PURPOSE is unenforceable by an end-user for most (if not all) NDcPP targets because it assumes a user can modify the TOE. **OE.NO_GENERAL_PURPOSE** is in effect because the TOE is not provided with general-purpose computing capabilities. **OE.TRUSTED_ADMIN** is covered by an explicit statement in the CC Guide. **OE.UPDATES** is covered in the CC Guide under the **''Performing Secure Upgrade**' section in the CC Guide. **OE.ADMIN_CREDENTIALS_SECUR**E – The CC Guide, throughout all sections, the document directs administrators to protect their administrator access credentials, respectively. The Security Target, section 6 - FCS_CKM.4 describes the credential securing methods used. **OE.RESIDUAL_INFORMATION** is covered in the CC guide as it covers methods to zeroize the device back to factory default states.

OE.CONNECTIONS – the admin guide documents covers this in detail on the Magnum server usage and syslog server communication.

Verdict:
PASS.

6.2.2.2 (5.3.2.2) EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator checked the requirements above are met by the preparative procedures. The entire AGD was used to determine the verdict of this work unit. Upon investigation, the evaluator found that the AGD – section "**Introduction"** describes each of the devices in the operating environment, including: • **Syslog Server** • **Management Workstation** • **CRL Server** • **Evertz Magnum Server** • **Media Gateway (Optional)** • **Video Destination Devices (Optional)** • **Video Source Devices (Optional)**

Verdict:
PASS.

6.2.2.3 (5.3.2.3) EVALUATION ACTIVITY

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

Evaluator Findings:
The evaluator checked the requirements are met by the preparative procedures. The entire AGD was used to determine the verdict of this work unit. Upon investigation, the evaluator found that AGD describes all of the instructions necessary to install and configure the TOE to work in the target operating environment, including: • **Administer the TOE locally and remotely.** • **Configure the authentication failure parameters.** • **Update the Magnum, and verify the updates using digital signature capability prior to installing those updates.** • **Resetting passwords.** • **Administrative login and logout.** • **Generate CSRs, import x509 certificates, and delete x509 certificates.** • **Configure the access banner.** • **Configure the session inactivity time before session termination or locking.**

- **Configure remote audit server parameters.**
- **Set the time which is used for time-stamps.**

Verdict:
PASS.

6.2.2.4 (5.3.2.4) EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator ensured the preparative procedures include instructions to manage the security of the TSF as a product and as a component of the larger operational environment. The entire AGD was used to determine the verdict of this work unit. The same commands, configurations, and interfaces used to install the TOE are also used for ongoing management, so this is satisfied by AGD_PRE.1 Test #3

Verdict:
PASS.

6.2.2.5 (5.3.2.5) EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator ensured the preparative procedures include instructions to provide a protected administrative capability and changing default passwords. The sections titled **"Configure Access Controls"** were used to determine the verdict of this work unit. The AGD describes changing the default password associated with the root account Based on these findings, this assurance activity is considered satisfied.

Verdict:
PASS.

6.3 AVA: VULNERABILITY ASSESSMENT

6.3.1 VULNERABILITY SURVEY (AVA_VAN.1)

6.3.1.1 (5.6.1.1) EVALUATION ACTIVITY (DOCUMENTATION) **[TD0547]**

In addition to the activities specified by the CEM in accordance with Table 2, the evaluator shall perform the following activities.

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 developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components should identify at a minimum the processors used by the TOE. Software components include applications, the operating system and other major components that are independently identifiable and reusable (outside of the TOE), for example a web server, protocol or cryptographic libraries, (independently identifiable and reusable components are not limited to the list provided in the example). This additional documentation is merely a list of the name and version number of the components and will be used by the evaluators in formulating vulnerability hypotheses during their analysis.
If the TOE is a distributed TOE then the developer shall provide:

a. documentation describing the allocation of requirements between distributed TOE components as in [NDcPP, 3.4]
b. a mapping of the auditable events recorded by each distributed TOE component as in [NDcPP, 6.3.3]
c. additional information in the Preparative Procedures as identified in the refinement of AGD_PRE.1 in additional information in the Preparative Procedures as identified in 3.4.1.2 and 3.5.1.2.

Evaluator Findings:
The evaluator collected this information from the developer which was used to feed into the Public Domain Search. Refer to evaluator findings in the evaluation activity below.

Verdict:
PASS.

6.3.1.2 (5.6.1.2) EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator documented their analysis and testing of potential vulnerabilities with respect to this requirement. Public searches were performed against all keywords found within the Security Target and AGD that may be applicable to specific TOE components. This included protocols, TOE software version, and TOE hardware to ensure sufficient coverage under AVA. The evaluator searched the Internet for potential vulnerabilities in the TOE using the web sites listed below. The sources of the publicly available information are provided below:

- http://nvd.nist.gov/
- http://www.us-cert.gov
- http://www.securityfocus.com/
- https://www.cvedetails.com/
- www.exploitsearch.net
- www.securiteam.com
- http://nessus.org/plugins/index.php?view=search
- http://www.zerodayinitiative.com/advisories
- https://www.exploit-db.com
- https://www.rapid7.com/db/vulnerabilities

The evaluator performed the public domain vulnerability searches using the following key words. The search was performed on 13th August 2024.

- **Evertz EXE**
- **Evertz**
- **Intel-Core i3 6102E**
- **rsyslogd 8.2010.0**
- **Lighttpd 1.4.59**
- **OpenSSL 1.1.1k**
- **Linux Kernel 4.19.165**

The evaluation lab examined each result provided from NVD and Exploit Search to determine if the current TOE version or component within the environment was vulnerable. Based upon the analysis, any issues found that were generated were patched in the TOE version and prior versions, mitigating the risk factor.

Verdict:
PASS.

7 DETAILED TEST CASES (TEST ACTIVITIES)

7.1 AUDIT

7.1.1 FAU_GEN.1 TEST #1

Item	Data
Test Assurance Activity	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. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.
Notes	**NIT Technical Decision (TD0639) for Clarification for NTP MAC Keys.** The SFRs FAU_GEN.1, FCS_CKM.4 and FPT_SKP_EXT.1 shall be applied to all cryptographic keys that are related to secure communication (i.e. related to FTP_TRP.1, FTP_ITC.1, FPT_ITT.1). The NTP requirements have been introduced in NDcPP V2.1 as a rather 'standalone' set of requirements with 'no audit requirements' specified in the ECD section for FCS_NTP_EXT.1 and no dependencies on FCS_CKM - in contrast to the corresponding sections for secure communication protocols like TLS. As NTP keys are not intended to be used for encryption of sensitive information, the level of protection is different compared to other pre-shared keys. It has therefore not been intended that NTP keys are treated as other pre-shared keys in the context of NDcPP.
Audit Log Requirement	[PP] FAU_GEN.1.1 audit requirements [PP] FAU_GEN.1.2, Application Note 3, Table 2 [PP] B.1 Audit Events for Selection-Based SFRs, table 5
Test Steps	• Trigger each auditable event on the TOE. • Verify that each audit record is generated and contains the required information.

Expected Test Results	• The TOE can generate audit records for each of the events described in the ST under the FAU_GEN.1.1 & 1.2 along with the events mentioned in Table 12 of the ST. • The TOE can generate audit records for establishment and termination of a channel for HTTPS/TLS. • The audit records generated match the proper format as specified in the guidance documentation.
Pass/Fail with Explanation	Pass. Covered by audit records in each test case. This meets the testing requirements.

7.1.2 FAU_GEN.1 TEST #2A

Item	Data
Test Assurance Activity	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.
Notes	**NIT Technical Decision (TD0639) for Clarification for NTP MAC Keys.** The SFRs FAU_GEN.1, FCS_CKM.4 and FPT_SKP_EXT.1 shall be applied to all cryptographic keys that are related to secure communication (i.e. related to FTP_TRP.1, FTP_ITC.1, FPT_ITT.1). The NTP requirements have been introduced in NDcPP V2.1 as a rather 'standalone' set of requirements with 'no audit requirements' specified in the ECD section for FCS_NTP_EXT.1 and no dependencies on FCS_CKM - in contrast to the corresponding sections for secure communication protocols like TLS. As NTP keys are not intended to be used for encryption of sensitive information, the level of protection is different compared to other pre-shared keys. It has therefore not been intended that NTP keys are treated as other pre-shared keys in the context of NDcPP.

Pass/Fail with Explanation	NA. This TOE is not a distributed TOE.

7.1.3 FAU_GEN.2 TEST #2B

Item	Data
Test Assurance Activity	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.
Notes	**NIT Technical Decision (TD0639) for Clarification for NTP MAC Keys.** The SFRs FAU_GEN.1, FCS_CKM.4 and FPT_SKP_EXT.1 shall be applied to all cryptographic keys that are related to secure communication (i.e. related to FTP_TRP.1, FTP_ITC.1, FPT_ITT.1). The NTP requirements have been introduced in NDcPP V2.1 as a rather 'standalone' set of requirements with 'no audit requirements' specified in the ECD section for FCS_NTP_EXT.1 and no dependencies on FCS_CKM - in contrast to the corresponding sections for secure communication protocols like TLS. As NTP keys are not intended to be used for encryption of sensitive information, the level of protection is different compared to other pre-shared keys. It has therefore not been intended that NTP keys are treated as other pre-shared keys in the context of NDcPP.
Pass/Fail with Explanation	NA. This TOE is not a distributed TOE.

7.1.4 FAU_STG_EXT.1 TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FAU_STG_EXT.1: No audit requirements for FAU_STG.1. [PP] FMT_SMF.1: If the characteristics of the remote log settings are changed, then these changes need to be audited.
Test Steps	• Configure the TOE to send logs to a syslog server. • Verify that the audit logs for configuration of the audit log setting were audited. • Restart the syslog service on Syslog Server. • Verify the syslog version on VM. • Login to the TOE to generate logs. • Verify with the TOE's local logs that there are audit logs for user logged in successfully. • Verify that the logs are seen on the remote syslog server are the same as on TOE. • Verify with packet capture that the logs that are sent to the remote syslog server are encrypted.
Expected Test Results	Screenshots showing that logs generated on the TOE are the same as those transferred to the external audit server. Packet capture showing that logs sent to the external audit server are encrypted.
Pass/Fail with Explanation	Pass. TOE can transfer audit data in encrypted format to an external audit server. This meets the requirement.

7.1.5 FAU_STG_EXT.1 TEST #2 (A)

Item	Data
Test Assurance Activity	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:

	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).
Pass/Fail with Explanation	NA. The TOE overwrites the previous audit records when the local storage space for audit data is full.

7.1.6 FAU_STG_EXT.1 TEST #2 (B)

Item	Data
Test Assurance Activity	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: 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)
Audit Log Requirement	[PP] FAU_STG_EXT.1: No audit requirements for FAU_STG.1.
Test Steps	• Verify oldest log in /var/log/messages file. • Generate dummy logs to fill /var/log/messages file size to 100%. • Verify logs again and old logs were overwritten with new logs.
Expected Test Results	The TOE should successfully allow the overwriting of old logs by new ones.
Pass/Fail with Explanation	Pass. The test is passed because once the limit was reached the oldest audit record was overwritten. This meets the testing requirements.

7.1.7 FAU_STG_EXT.1 TEST #2 (C)

Item	Data
Test Assurance Activity	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: The TOE behaves as specified (for the option '**other action**' in FAU_STG_EXT.1.3).
Pass/Fail with Explanation	NA. The TOE does not claim any other action other than overwriting existing logs.

7.1.8 FAU_STG_EXT.1 TEST #3

Item	Data
Test Assurance Activity	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
Pass/Fail with Explanation	NA. This TOE is not a distributed TOE.

7.1.9 FAU_STG_EXT.1 TEST #4

Item	Data
Test Assurance Activity	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.

Pass/Fail with Explanation	NA. This TOE is not a distributed TOE.

7.1.10 FPT_STM_EXT.1 TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FMT_SMF.1: Changes to the time should be audited. The audit log should contain the old and the new times.
Test Steps	• Confirm the current time on the TOE. • Set a new time on the TOE via the local console. • Verify that there is an audit log for the successful time set.
Expected Test Results	• The TOE allows time to be set manually via local console using the 'Set time' option and via Console. This can be seen in screenshots showing the time on the TOE being updated via local console. • Audit logs also show the TOE time being modified manually via local console.
Pass/Fail with Explanation	Pass. The TOE allows the administrative user to configure the time on the TOE. This meets the testing requirements.

7.1.11 FPT_STM_EXT.1 TEST #2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	NA. The TOE does not claim NTP and does not support independent time information. Hence this test is not applicable.

7.1.12 FPT_STM_EXT.1 TEST #3 [TD0632]

Item	Data
Test Assurance Activity	[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. **TD0632 has been applied.**
Pass/Fail with Explanation	NA. TOE does not "obtain time from the underlying virtualization system".

7.1.13 FTP_ITC.1 TEST #1

Item	Data
Test Assurance Activity	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.
Notes	**The NiT has issued a technical decision (TD0572) for Restricting FTP_ITC.1 to only IP address identifiers.**

	DNS resolution is not mandatory to support FTP_ITC.1. There are no resolution mandates or requirements in RFC 6125. The use of the dnsName identifiers in X.509 certificates must be supported by TOEs that claim FCS_DTLSC_EXT or FCS_TLSC_EXT when associated with FTP_ITC.1. Inability to parse dnsName identifiers shall be considered a failure to meet these requirements.
Audit Log Requirement	[PP] FTP_ITC.1: Initiation of the trusted channel should be audited.
Test Steps	This test was performed in conjunction with FAU_STG_EXT.1 Test #1 for Syslog channel and FPT_ITC.1 Test #4 for both syslog channel and the Synergy (Magnum) Channel. As that test showed all communications with an external syslog server and a Magnum Server are protected by TLS encryption.
Expected Test Result	This test was performed in conjunction with FAU_STG_EXT.1 Test #1 for Syslog channel and FPT_ITC.1 Test #4 for both syslog channel and the Synergy (Magnum) Channel. As that test showed all communications with an external syslog server and a Magnum Server are protected by TLS encryption.
Pass/Fail with Explanation	Pass. This test was performed in conjunction with FAU_STG_EXT.1 Test #1 for Syslog channel and FPT_ITC.1 Test #4 for both syslog channel and the Synergy (Magnum) Channel. As that test showed all communications with an external syslog server and a Magnum Server are protected by TLS encryption.

7.1.14 FTP_ITC.1 TEST #2

Item	Data
Test Assurance Activity	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.
Notes	**The NiT has issued a technical decision (TD0572) for Restricting FTP_ITC.1 to only IP address identifiers.** DNS resolution is not mandatory to support FTP_ITC.1. There are no resolution mandates or requirements in RFC 6125.

	The use of the dnsName identifiers in X.509 certificates must be supported by TOEs that claim FCS_DTLSC_EXT or FCS_TLSC_EXT when associated with FTP_ITC.1. Inability to parse dnsName identifiers shall be considered a failure to meet these requirements.
Audit Log Requirement	[PP] FTP_ITC.1: Initiation of the trusted channel should be audited.
Test Steps	This test was performed in conjunction with FAU_STG_EXT.1 Test #1 and FPT_ITC.1 Test #4. The PCAPs shows that it is the TOE (10.1.5.5) responsible for initiating the TCP SYN 3-way handshake. It then sets up the TLS handshake by transmitting the TLS Client Hello packet. The Magnum Server is responsible for initiating the TCP SYN 3-way handshake for the Synergy Channel.
Expected Test Results	This test was performed in conjunction with FAU_STG_EXT.1 Test #1 and FPT_ITC.1 Test #4. The PCAPs shows that it is the TOE (10.1.5.5) responsible for initiating the TCP SYN 3-way handshake. It then sets up the TLS handshake by transmitting the TLS Client Hello packet. The Magnum Server (10.1.5.44) is responsible for initiating the TCP SYN 3-way handshake for the Synergy Channel.
Pass/Fail with Explanation	Pass. This test was performed in conjunction with FAU_STG_EXT.1 Test #1 and FPT_ITC.1 Test #4. The PCAPs shows that it is the TOE (10.1.5.5) responsible for initiating the TCP SYN 3-way handshake. It then sets up the TLS handshake by transmitting the TLS Client Hello packet. The Magnum Server is responsible for initiating the TCP SYN 3-way handshake for the Synergy Channel.

7.1.15 FTP_ITC.1 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
Notes	**The NiT has issued a technical decision (TD0572) for Restricting FTP_ITC.1 to only IP address identifiers.** DNS resolution is not mandatory to support FTP_ITC.1. There are no resolution mandates or requirements in RFC 6125.

	The use of the dnsName identifiers in X.509 certificates must be supported by TOEs that claim FCS_DTLSC_EXT or FCS_TLSC_EXT when associated with FTP_ITC.1. Inability to parse dnsName identifiers shall be considered a failure to meet these requirements.
Audit Log Requirement	[PP] FTP_ITC.1: Initiation of the trusted channel should be audited. [PP] FMT_SMF.1: When trusted channel path are managed.
Test Steps	This test was covered while conducting FAU_STG_EXT.1 Test #1 for Syslog Channel and FPT_ITC.1 Test #4 for Synergy (Magnum) Channel. As that test showed all communications with an external syslog server and the Magnum Server are protected by TLS encryption. For both claimed channels, the traffic was observed to be encrypted.
Expected Test Results	While making a connection between TOE and IT entity (Syslog Server and the Magnum Server), traffic should traverse in encrypted format (TLS Encryption) between these two devices.
Pass/Fail with Explanation	Pass. This test was performed in conjunction with FAU_STG_EXT.1 Test #1 and FPT_ITC.1 Test #4. As that test showed, all communications with an external syslog server and Magnum Server are protected by TLS encryption.

7.1.16 FTP_ITC.1 TEST #4

Item	Data
Test Assurance Activity	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: 1. A duration that exceeds the TOE's application layer timeout setting, 2. 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.
Notes	**The NiT has issued a technical decision (TD0572) for Restricting FTP_ITC.1 to only IP address identifiers.** DNS resolution is not mandatory to support FTP_ITC.1. There are no resolution mandates or requirements in RFC 6125. The use of the dnsName identifiers in X.509 certificates must be supported by TOEs that claim FCS_DTLSC_EXT or FCS_TLSC_EXT when associated with FTP_ITC.1. Inability to parse dnsName identifiers shall be considered a failure to meet these requirements.
Audit Log Requirement	[PP] FTP_ITC.1: • Initiation of the trusted channel • Termination of the trusted channel • Failure of the trusted channel with information of the initiator and the target
Test Steps	**For Syslog.** Short Disconnect • Initiate a connection with the remote server. • Jack-In/Jack-Out LAN cable with remote server for short period of time – Before TOE Application gets timed-out. • Verify with packet capture that session is disconnected and restored with no TSF data is sent in plaintext. Long Disconnect • Initiate a connection with the remote server. • Jack-In/Jack-Out LAN cable with remote server for long period of time – Till TOE Application gets timed-out. • Verify with packet capture that session is disconnected and restored with no TSF data is sent in plaintext. **For Synergy.** Short Disconnect • Initiate a connection with the remote server. • Jack-In/Jack-Out LAN cable with remote server for short period of time – Before TOE Application gets timed-out.

	• Verify with packet capture that session is disconnected and restored with no TSF data is sent in plaintext. Long Disconnect • Initiate a connection with the remote server. • Jack-In/Jack-Out LAN cable with remote server for long period of time – Till TOE Application gets timed-out. • Verify with packet capture that session is disconnected and restored with no TSF data is sent in plaintext.
Expected Test Results	When physical connectivity with a remote audit server is interrupted and then restored, the data is exchanged between both entities is never in plaintext, as can be shown by packet captures during and after the outage.
Pass/Fail with Explanation	Pass. The TOE does not send plaintext traffic when disconnected from the log server. This meets the testing requirements.

7.2 AUTH

7.2.1 FCS_CKM.1 RSA

Item	Data
Test Assurance Activity	Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). **Key Generation for FIPS PUB 186-4 RSA Schemes** The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent *e*, the private prime factors *p* and *q*, the public modulus *n* and the calculation of the private signature exponent d. Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include: a) Random Primes: • Provable primes • Probable primes b) Primes with Conditions: • Primes p1, p2, q1, q2, p and q shall all be provable primes • Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes • Primes p1, p2, q1, q2, p and q shall all be probable primes To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation.
Pass/Fail with Explanation	Algorithm: RSA KeyGen

	Key size / Modulus: 2048-bits, 3072-bits, and 4096-bits CAVP #: A2573 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.2 FCS_CKM.1 ECC

Item	Data
Test Assurance Activity	Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). **Key Generation for Elliptic Curve Cryptography (ECC)** *FIPS 186-4 ECC Key Generation Test* For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation. *FIPS 186-4 Public Key Verification (PKV) Test* For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.
Pass/Fail with Explanation	Algorithm: ECDSA KeyGen and ECDSA KeyVer Curves: P-256, P-384, P-521

	CAVP #: A2573 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.3 FCS_CKM.1 FFC – FIPS PUB 186-4

Item	Data
Test Assurance Activity	Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). **Key Generation for Finite-Field Cryptography (FFC)** The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the TOE using the Parameter Generation and Key Generation test. This test verifies the ability of the TSF to correctly produce values for the field prime p, the cryptographic prime q (dividing p-1), the cryptographic group generator g, and the calculation of the private key x and public key y. The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p: • Primes q and p shall both be provable primes • Primes q and field prime p shall both be probable primes and two ways to generate the cryptographic group generator g: • Generator g constructed through a verifiable process • Generator g constructed through an unverifiable process. The Key generation specifies 2 ways to generate the private key x: • len(q) bit output of RBG where 1 <=x <= q-1

	• len(q) + 64 bit output of RBG, followed by a mod q-1 operation and a +1 operation, where 1<= x<=q-1. The security strength of the RBG must be at least that of the security offered by the FFC parameter set. To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set. For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm • g != 0,1 • q divides p-1 • g^q mod p = 1 • g^x mod p = y for each FFC parameter set and key pair.
Pass/Fail with Explanation	NA. This scheme is not selected in FCS_CKM.1.1 SFR in ST. Hence, this test is not applicable for this TOE.

7.2.4 FCS_CKM.1 FFC – "SAFE-PRIME" GROUPS

Item	Data
Test Assurance Activity	Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported).

	FFC Schemes using "safe-prime" groups Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1. **TD0580 has been applied.**
Pass/Fail with Explanation	NA. This scheme is not selected in FCS_CKM.1.1 SFR in ST. Hence, this test is not applicable for this TOE.

7.2.5 FCS_CKM.2 RSA

Item	Data
Test Assurance Activity	**RSA-based key establishment** The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 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 RSAES-PKCS1-v1_5.
Pass/Fail with Explanation	Pass. This testing was performed in conjunction with FTP_TRP.1/Admin Test #1 and FTP_ITC.1 Test #1 to demonstrate correct operation.

7.2.6 FCS_CKM.2 SP800-56A - ECC

Item	Data
Test Assurance Activity	**Key Establishment Schemes** The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE using the applicable tests below.

SP800-56A Key Establishment Schemes

The evaluator shall verify a TOE's implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag.

Function Test

The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested.

The evaluator shall obtain the DKM, the corresponding TOE's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields.

If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret.

The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values.

If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm.

	Validity Test The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOE's public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE id fields. The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOE's static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass). The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results with the results using a known good implementation verifying that the TOE detects these errors.
Pass/Fail with Explanation	Algorithm: KAS-ECC Sp800-56Ar3 (or KAS-ECC-SSC Sp800-56Ar3) CAVP #: A2573 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.7 FCS_CKM.2 SP800-56A - FFC

Item	Data
Test Assurance Activity	**Key Establishment Schemes** The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE using the applicable tests below. **SP800-56A Key Establishment Schemes** The evaluator shall verify a TOE's implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag. *Function Test* The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested. The evaluator shall obtain the DKM, the corresponding TOE's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields. If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret.

	The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values. If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm. *Validity Test* The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOE's public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE id fields. The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOE's static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass). The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results with the results using a known good implementation verifying that the TOE detects these errors.
Pass/Fail with Explanation	NA. This scheme is not selected in FCS_CKM.2.1 SFR in ST. Hence, this test is not applicable for this TOE.

7.2.8 FCS_CKM.2 FCC SAFE-PRIME

Item	Data
Test Assurance Activity	**FFC Schemes using "safe-prime" groups** 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.
Pass/Fail with Explanation	NA. This scheme is not selected in FCS_CKM.2.1 SFR in ST. Hence, this test is not applicable for this TOE.

7.2.9 FCS_CKM.4

Item	Data
Test Assurance Activity	There are no test assurance activities.
Notes	**NIT Technical Decision (TD0639) for Clarification for NTP MAC Keys.** The SFRs FAU_GEN.1, FCS_CKM.4 and FPT_SKP_EXT.1 shall be applied to all cryptographic keys that are related to secure communication (i.e. related to FTP_TRP.1, FTP_ITC.1, FPT_ITT.1). The NTP requirements have been introduced in NDcPP V2.1 as a rather 'standalone' set of requirements with 'no audit requirements' specified in the ECD section for FCS_NTP_EXT.1 and no dependencies on FCS_CKM - in contrast to the corresponding sections for secure communication protocols like TLS. As NTP keys are not intended to be used for encryption of sensitive information, the level of protection is different compared to other pre-shared keys. It has therefore not been intended that NTP keys are treated as other pre-shared keys in the context of NDcPP.
Pass/Fail with Explanation	NA. There are no test assurance activities for this SFR. Based on these findings, this assurance activity is considered satisfied.

7.2.10 FCS_COP.1/DATAENCRYPTION AES-CBC

Item	Data
Test Assurance Activity	**AES-CBC Known Answer Tests** There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation. **KAT-1.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key. To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption. **KAT-2.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys. To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AESCBC decryption. **KAT-3**. To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N].

	To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of keys and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key. **KAT-4.** To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256- bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128]. To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption. **AES-CBC Multi-Block Message Test** The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation. The evaluator shall also test the decrypt functionality for each mode by decrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation. **AES-CBC Monte Carlo Tests**

	The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:
	
	# Input: PT, IV, Key
	
	for i = 1 to 1000:
	
	if i == 1:
	
	CT[1] = AES-CBC-Encrypt(Key, IV, PT)
	
	PT = IV
	
	else:
	
	CT[i] = AES-CBC-Encrypt(Key, PT)
	
	PT = CT[i-1]
	
	The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.
	
	The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AESCBC-Decrypt.

Pass/Fail with Explanation	Algorithm: AES CBC Key size: 128 bits, 256 bits CAVP #: A2573 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.11 FCS_COP.1/DATAENCRYPTION AES-GCM

Item	Data
Test Assurance Activity	**AES-GCM Test** The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths: **128 bit and 256 bit keys** a) **Two plaintext lengths.** One of the plaintext lengths shall be a nonzero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported. a) **Three AAD lengths.** One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported. b) **Two IV lengths.** If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested. The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known. The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.
Pass/Fail with Explanation	Algorithm: AES GCM Key size: 128 bits, 256 bits CAVP #: A2573

	Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.12 FCS_COP.1/DATAENCRYPTION AES-CTR

Item	Data
Test Assurance Activity	**AES-CTR Known Answer Tests** The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. Since the Counter Mode does not specify the counter that is used, it is not possible to implement an automated test for this mode. The generation and management of the counter is tested through FCS_SSH*_EXT.1.4. If CBC and/or GCM are selected in FCS_COP.1/DataEncryption, the test activities for those modes sufficiently demonstrate the correctness of the AES algorithm. If CTR is the only selection in FCS_COP.1/DataEncryption, the AES-CBC Known Answer Test, AESGCM Known Answer Test, or the following test shall be performed (all of these tests demonstrate the correctness of the AES algorithm): There are four Known Answer Tests (KATs) described below to test a basic AES encryption operation (AES-ECB mode). For all KATs, the plaintext, IV, and ciphertext values shall be 128-bit blocks. The results from each test may either be obtained by the validator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation. KAT-1 To test the encrypt functionality, the evaluator shall supply a set of 5 plaintext values for each selected keysize and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros. KAT-2 To test the encrypt functionality, the evaluator shall supply a set of 5 key values for each selected keysize and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value.

KAT-3 To test the encrypt functionality, the evaluator shall supply a set of key values for each selected keysize as described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values. A set of 128 128-bit keys, a set of 192 192-bit keys, and/or a set of 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N].

KAT-4 To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the ciphertext values that result from encryption of the given plaintext using each selected keysize with a key value of all zeros (e.g. 256 ciphertext values will be generated if 128 bits and 256 bits are selected and 384 ciphertext values will be generated if all keysizes are selected). Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128].

AES-CTR Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10 (test shall be performed using AES-ECB mode). For each i the evaluator shall choose a key and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key using a known good implementation. The evaluator shall perform this test using each selected keysize.

AES-CTR Monte-Carlo Test

The evaluator shall test the encrypt functionality using 100 plaintext/key pairs. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows:

Input: PT, Key

for i = 1 to 1000:

CT[i] = AES-ECB-Encrypt(Key, PT) PT = CT[i]

The ciphertext computed in the 1000th iteration is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation. The evaluator shall perform this test using each selected keysize.

There is no need to test the decryption engine.

Pass/Fail with Explanation	Algorithm: AES CTR Key size: 128 bits, 256 bits CAVP #: A2573 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.13 FCS_COP.1/SIGGEN ECDSA

Item	Data
Test Assurance Activity	**ECDSA Algorithm Tests** **ECDSA FIPS 186-4 Signature Generation Test** For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation. **ECDSA FIPS 186-4 Signature Verification Test** For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.
Pass/Fail with Explanation	NA. This scheme is not selected in FCS_COP.1.1/SigGen SFR in ST. Hence, this test is not applicable for this TOE.

7.2.14 FCS_COP.1/SIGGEN RSA

Item	Data
Test Assurance Activity	**RSA Signature Algorithm Tests** **Signature Generation Test** The evaluator generates or obtains 10 messages for each modulus size/SHA combination supported by the TOE. The TOE generates and returns the corresponding signatures. The evaluator shall verify the correctness of the TOE's signature using a trusted reference implementation of the signature verification algorithm and the associated public keys to verify the signatures. **Signature Verification Test** For each modulus size/hash algorithm selected, the evaluator generates a modulus and three associated key pairs, (d, e). Each private key d is used to sign six pseudorandom messages each of 1024 bits using a trusted reference implementation of the signature generation algorithm. Some of the public keys, e, messages, or signatures are altered so that signature verification should fail. For both the set of original messages and the set of altered messages: the modulus, hash algorithm, public key e values, messages, and signatures are forwarded to the TOE, which then attempts to verify the signatures and returns the verification results. The evaluator verifies that the TOE confirms correct signatures on the original messages and detects the errors introduced in the altered messages.
Pass/Fail with Explanation	Algorithm: RSA SigGen, RSA SigVer Key size / Modulus: 2048 bits, 3072 bits, 4096-bits CAVP #: A2573 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.15 FCS_COP.1/HASH

Item	Data
Test Assurance Activity	The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented testmacs. The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP. **Short Messages Test - Bit-oriented Mode** The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Short Messages Test - Byte-oriented Mode** The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Selected Long Messages Test - Bit-oriented Mode** The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

	Selected Long Messages Test - Byte-oriented Mode The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Pseudorandomly Generated Messages Test** This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.
Pass/Fail with Explanation	Algorithm: SHA-1, SHA-256, SHA-384 CAVP #: A2573 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.16 FCS_COP.1/KEYEDHASH

Item	Data
Test Assurance Activity	For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and message data using a known good implementation.
Pass/Fail with Explanation	Algorithm: HMAC (HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384) CAVP #: A2573

	Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.17 FCS_RBG_EXT.1

Item	Data
Test Assurance Activity	The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A). If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call. The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator. **Entropy input:** the length of the entropy input value must equal the seed length. **Nonce:** If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length.

	Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied. **Additional input:** the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.
Pass/Fail with Explanation	Algorithm: Counter DRBG Mode: CTR_DRBG (AES) CAVP #: A2573 Pass. Based on these findings, this assurance activity is considered satisfied.

7.2.18 FCS_HTTPS_EXT.1 TEST #1

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. The TOE acts as an HTTPS Server only for WebGUI access. The TOE acts as a TLS Server with mutual authentication and for both HTTPS and TLSS with mutual authentication, the same set of certificates is used, hence, for both channels, the certificate validity is done the same way. The certificate validity testing is covered with FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, and FIA_X509_EXT.1/Rev testing.

7.2.19 FIA_AFL.1 TEST #1

Item	Data
Test Assurance Activity	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.
Notes	**The NiT has issued a technical decision (TD0570) for clarification about FIA_AFL.1.** 1. FIA_AFL.1 is a mandatory SFRs that the TOE will need to meet. 2. FIA_AFL.1 requires at least one remote administrative interface support password authentication. 3. If SSH is the TOE's only remote administrative interface, it needs to support password authentication. If there is another administrative interface (e.g. a web GUI) that supports password authentication, SSH does not need to support password authentication and, by extension, FIA_AFL.1. **The NiT has issued a technical decision (TD0571) for Guidance on how to handle FIA_AFL.1.** 1. FIA_UAU_EXT.2.1 applies solely to how the administrator logs in at the local console. Passwords were made selection based to allow TOEs that have difficulty determining whether an incoming connection is local or remote to provide a mechanism to prevent administrative lockout. As FIA_AFL.1 is a mandatory SFR, it is expected that the TOE provides at least one remote password-based authentication mechanism using credentials managed by the TOE. 2. FIA_PMG_EXT.1, FIA_AFL.1, and FMT_SMF.1 are all mandatory SFRs that the TOE will need to meet.
Audit Log Requirement	[PP] FIA_AFL.1: When unsuccessful login attempts limit is met or exceeded. Audit log should show the origin of the attempt. [PP] FMT_SMF.1: Configuration of the authentication failure parameters for FIA_AFL.1
Test Steps	GUI: • Configure a maximum number (Three) of unsuccessful user authentication attempts to login into TOE. • Attempt to login three times to lock the user account with incorrect credentials & verify that it's locked. • Login with correct credentials and verify that it is not successful.

	• Verify that the failure of the user authentication is audited and the reason for failure is reflected in the audit log. Console: • Verify the console does not get locked out
Expected Test Results	Once configured maximum three number of unsuccessful authentication attempts on TOE, it will give user notification message (reject) for wrong credentials while login on to the TOE and in fourth attempt account will get locked out for the same user on the TOE.
Pass/Fail with Explanation	Pass. The TOE denied access to accounts after invalid authentication attempts and account getting locked out. This meets testing requirements.

7.2.20 FIA_AFL.1 TEST #2A

Item	Data
Test Assurance Activity	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).
Notes	**The NiT has issued a technical decision (TD0570) for clarification about FIA_AFL.1.** 1. FIA_AFL.1 is a mandatory SFRs that the TOE will need to meet. 2. FIA_AFL.1 requires at least one remote administrative interface support password authentication. 3. If SSH is the TOE's only remote administrative interface, it needs to support password authentication. If there is another administrative interface (e.g. a web GUI) that supports password authentication, SSH does not need to support password authentication and, by extension, FIA_AFL.1. **The NiT has issued a technical decision (TD0571) for Guidance on how to handle FIA_AFL.1.** 1. FIA_UAU_EXT.2.1 applies solely to how the administrator logs in at the local console. Passwords were made selection based to allow TOEs that have difficulty determining whether an incoming connection is local or remote to provide a mechanism to prevent administrative lockout. As FIA_AFL.1 is a mandatory SFR, it is

	expected that the TOE provides at least one remote password-based authentication mechanism using credentials managed by the TOE. 2. FIA_PMG_EXT.1, FIA_AFL.1, and FMT_SMF.1 are all mandatory SFRs that the TOE will need to meet.
Audit Log Requirement	[PP] FIA_AFL.1: When unsuccessful login attempts limit is met or exceeded. Audit log should show the origin of the attempt. [PP] FMT_SMF.1: Configuration of the authentication failure parameters for FIA_AFL.1
Test Steps	GUI: • Attempt to login to the TOE with incorrect credentials. • Verify that the user account is locked out, and the reason is reflected in the audit log. • Manually unlock the user account by Admin Account. • Verify that the user account is unlocked, and the reason is reflected in the audit log. • Login with good credentials. • Verify that the successful user login attempt audited and reflected in the audit log.
Expected Test Results	By making login attempt with wrong credentials, user account should get locked out and once locked user account unlocked by Admin user account then user can make successful login attempt to the TOE using his correct credentials.
Pass/Fail with Explanation	Pass. By making login attempts with wrong credentials, user account got locked out and post unlocking this account by Admin account, user was successfully able to make login attempt using his correct login credentials on the TOE. This meets the testing requirements.

7.2.21 FIA_AFL.1 TEST #2B

Item	Data
Test Assurance Activity	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 **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 authorization 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 authorization attempt using valid credentials results in successful access.
Notes	**The NiT has issued a technical decision (TD0570) for clarification about FIA_AFL.1.** 1. FIA_AFL.1 is a mandatory SFRs that the TOE will need to meet. 2. FIA_AFL.1 requires at least one remote administrative interface support password authentication. 3. If SSH is the TOE's only remote administrative interface, it needs to support password authentication. If there is another administrative interface (e.g. a web GUI) that supports password authentication, SSH does not need to support password authentication and, by extension, FIA_AFL.1. **The NiT has issued a technical decision (TD0571) for Guidance on how to handle FIA_AFL.1.** 1. FIA_UAU_EXT.2.1 applies solely to how the administrator logs in at the local console. Passwords were made selection based to allow TOEs that have difficulty determining whether an incoming connection is local or remote to provide a mechanism to prevent administrative lockout. As FIA_AFL.1 is a mandatory SFR, it is expected that the TOE provides at least one remote password-based authentication mechanism using credentials managed by the TOE. 2. FIA_PMG_EXT.1, FIA_AFL.1, and FMT_SMF.1 are all mandatory SFRs that the TOE will need to meet.
Pass/Fail with Explanation	NA. This functionality not claimed in ST. Hence, this test is not applicable for this TOE.

7.2.22 FIA_PMG_EXT.1 TEST #1

Item	Data
Test Assurance Activity	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.
Notes	**The NiT has issued a technical decision (TD0571) for Guidance on how to handle FIA_AFL.1.** 1. FIA_UAU_EXT.2.1 applies solely to how the administrator logs in at the local console. Passwords were made selection based to allow TOEs that have difficulty determining whether an incoming connection is local or remote to provide a mechanism to prevent administrative lockout. As FIA_AFL.1 is a mandatory SFR, it is expected that the TOE provides at least one remote password-based authentication mechanism using credentials managed by the TOE. 2. FIA_PMG_EXT.1, FIA_AFL.1, and FMT_SMF.1 are all mandatory SFRs that the TOE will need to meet.

Audit Log Requirement	[PP] FIA_PMG_EXT.1: There are no audit requirements for FIA_PMG_EXT.1 [PP] FAU_GEN.1: Resetting of passwords [PP] FAU_GEN.1: Changes to TSF data related to configuration changes (all changes that results in an update to the configuration file should be audited)
Test Steps	• Configure TOE for strong password practices according to the NDcPPv2.2e compliance in the ST. • Ensure that the default minimum password length of 15 characters long is set by default on TOE device. GUI: • Create username: good_user1 password: ABCD1234!@#abcd • Verify the successful creation of the user (good_user1) message reflected in audit log. • Create username: good_user2 password: EFGH5678$%^efgh • Verify the successful creation of the user (good_user2) message reflected in audit log. • Create username: good_user3 password: IJKL9012&*(ijkl • Verify the successful creation of the user (good_user3) message reflected in audit log. • Create username: good_user4 password: MNOP3456)!@mnop • Verify the successful creation of the user (good_user4) message reflected in audit log.
Expected Test Results	The TOE accepts valid password combinations that meet the requirements on GUI. Audit logs show that the user with the valid password combination has been added successfully.
Pass/Fail with Explanation	Pass. The TOE was able to create users with good passwords. This meets the testing requirements.

7.2.23 FIA_PMG_EXT.1 TEST #2

Item	Data
Test Assurance Activity	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.
Notes	**The NiT has issued a technical decision (TD0571) for Guidance on how to handle FIA_AFL.1.** 1. FIA_UAU_EXT.2.1 applies solely to how the administrator logs in at the local console. Passwords were made selection based to allow TOEs that have difficulty determining whether an incoming connection is local or remote to provide a mechanism to prevent administrative lockout. As FIA_AFL.1 is a mandatory SFR, it is expected that the TOE provides at least one remote password-based authentication mechanism using credentials managed by the TOE. 2. FIA_PMG_EXT.1, FIA_AFL.1, and FMT_SMF.1 are all mandatory SFRs that the TOE will need to meet.
Audit Log Requirement	[PP] FIA_PMG_EXT.1: There are no audit requirements for FIA_PMG_EXT.1 [PP] FAU_GEN.1: Resetting of passwords [PP] FAU_GEN.1: Changes to TSF data related to configuration changes (all changes that results in an update to the configuration file should be audited)
Test Steps	• Configure TOE for strong password practices according to the NDcPPv2.2e compliance in the ST. GUI: • Create username: “bad_user1” password: abcde!!!12345678 • Try to set a password (Lack of Upper-Case Letters) which does not meet the password complexity requirement. • Verify that the failure of the user (bad_user1) creation is audited and the reason for failure is reflected in the audit log. • Create username: “bad_user2” password: IJKLM@@@ijklmno • Try to set a password (Lack of Numbers) which does not meet the password complexity requirement. • Verify that the failure of the user (bad_user2) creation is audited and the reason for failure is reflected in the audit log. • Create username: “bad_user3” password: qrstuvWXYZ12345 • Try to set a password (Lack of Special Characters) which does not meet the password complexity requirement. • Verify that the failure of the user (bad_user3) creation is audited and the reason for failure is reflected in the audit log. • Create username: “bad_user4” password: ABCDE$@fgh123 • Try to set a password (Lack of Minimum Number of Characters) which does not meet the password complexity requirement. • Verify that the failure of the user (bad_user4) creation is audited and the reason for failure is reflected in the audit log. • Create username: “bad_user5” password: ABCDE!!!12345678 • Try to set a password (Lack of Lower Case Letters) which does not meet the password complexity requirement. • Verify that the failure of the user (bad_user5) creation is audited and the reason for failure is reflected in the audit log.

	• Create username: “bad_user6” password: No Password • Try to set a password (No Password) which does not meet the password complexity requirement. • Verify that the failure of the user (bad_user6) creation is audited and the reason for failure is reflected in the audit log.
Expected Test Results	The TOE only accepts valid password combinations on remote GUI. Audit logs show that addition of users with bad password combinations result in failure due to password did not meet “Password Complexity Criteria”.
Pass/Fail with Explanation	Pass. The TOE rejects user creation with bad passwords. This meets the testing requirements.

7.2.24 FIA_UIA_EXT.1 TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FIA_UIA_EXT.1: Successful and unsuccessful login attempts shall be audited. The audit log shall have the origin of the attempt
Test Steps	Console: • Attempt to login from a local connection with incorrect credentials. • Verify that the failure of the login is audited and the reason for failure is reflected in the audit log. • Log into the TOE from a local connection with correct credentials. • Verify that the login successfully message is audited and reflected in the audit log. GUI: • Attempt to login from a remote GUI connection with incorrect credentials. • Verify that the failure of the login is audited and the reason for failure is reflected in the audit log.

	• Log into the TOE from a remote GUI connection with correct credentials. • Verify that the login successfully message is audited and reflected in the audit log.
Expected Test Results	The TOE only allows an authorized user to gain access to the system via console and HTTPS. Users with incorrect credentials are denied access as shown by audit logs generated.
Pass/Fail with Explanation	Pass. Presenting incorrect authentication credentials results in being denied access to the TOE. Presenting correct authentication credentials results in access being allowed to the TOE. This meets the testing requirements.

7.2.25 FIA_UIA_EXT.1 TEST #2

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FIA_UIA_EXT.1: Successful and unsuccessful login attempts shall be audited. The audit log shall have the origin of the attempt.
Test Steps	GUI: • At the remote GUI, verify that no functionality except those specified in the requirement is allowed.
Expected Test Results	No services except displaying a banner is available to a remote administrator attempting to login to the TOE via GUI.
Pass/Fail with Explanation	Pass. No system services are available to an unauthenticated user connecting remotely. This meets the testing requirements.

7.2.26 FIA_UIA_EXT.1 TEST #3

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FIA_UIA_EXT.1: Successful and unsuccessful login attempts shall be audited. The audit log shall have the origin of the attempt
Test Steps	• At the directly connected console authentication prompt attempt to execute authenticated commands. • Verify that the no additional functionality is reflected in the audit log.
Expected Test Results	There are no services available to the user before authentication.
Pass/Fail with Explanation	Pass. There are no services available to the user before authentication. This meets testing requirements.

7.2.27 FIA_UIA_EXT.1 TEST #4

Item	Data
Test Assurance Activity	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.

Pass/Fail with Explanation	NA. The TOE is not a Distributed TOE. Hence, this test is not applicable.

7.2.28 FIA_UAU_EXT.2 TEST #1

Item	Data
Test Assurance Activity	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.
Notes	**The NiT has issued a technical decision (TD0571) for Guidance on how to handle FIA_AFL.1.** 1. FIA_UAU_EXT.2.1 applies solely to how the administrator logs in at the local console. Passwords were made selection based to allow TOEs that have difficulty determining whether an incoming connection is local or remote to provide a mechanism to prevent administrative lockout. As FIA_AFL.1 is a mandatory SFR, it is expected that the TOE provides at least one remote password-based authentication mechanism using credentials managed by the TOE. 2. FIA_PMG_EXT.1, FIA_AFL.1, and FMT_SMF.1 are all mandatory SFRs that the TOE will need to meet.
Pass/Fail with Explanation	Pass. Evaluation Activities for this requirement are covered under those for FIA_UIA_EXT.1.

7.2.29 FIA_UAU.7 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each method of local login allowed: 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.
Audit Log Requirement	There are no audit log requirements for this SFR.

Test Steps	• At the directly connected login prompt, enter authentication credentials. Verify that at most obscured feedback is provided.
Expected Test Results	The TOE should not provide anything other than obscured feedback, when entering the authentication information.
Pass/Fail with Explanation	Pass. The evaluator has verified that obscured feedback is provided while entering the authentication information. This meets the testing requirements.

7.2.30 FMT_MOF.1/MANUALUPDATE TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FMT_MOF.1/ManualUpdate: Attempt to initiate a manual update [PP] FAU_GEN.1: Changes to TSF data related to configuration changes (all changes that results in an update to the configuration file should be audited)
Test Steps	• Login with an unprivileged user account. • Attempt to upload firmware. • Verify that the failure of the manual update is audited and the reason for failure is reflected in the audit log.
Expected Test Results	An unprivileged user should not have the option to update a legitimate image
Pass/Fail with Explanation	Pass. Tried with an unprivileged user (ro_user) to upgrade firmware but the “Upgrade” option is not available for unprivileged users. This meets requirements.

7.2.31 FMT_MOF.1/MANUALUPDATE TEST #2

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FMT_MOF.1/ManualUpdate: Attempt to initiate a manual update
Test Steps	This test has been covered by FPT_TUD_EXT.1 test #1
Expected Test Results	This test has been covered by FPT_TUD_EXT.1 test #1
Pass/Fail with Explanation	Pass. This test has been covered by FPT_TUD_EXT.1 test #1

7.2.32 FMT_MOF.1/FUNCTIONS (1) TEST #1

Item	Data
Test Assurance Activity	Test 1 (if **'transmission of audit data to external IT entity'** is selected from the second selection together with **'modify the behaviour of'** in the first selection): The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters 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 modify the security related parameters 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.

Audit Log Requirement	[PP] FMT_MOF.1/Functions: There are no audit logs for FMT_MOF.1/Functions [PP] FMT_SMF.1: Audit log is required when the behaviour of the transmission of audit data to an external IT entity is modified.
Test Steps	• Login to the TOE as a user with no administrator privileges. • Attempt to modify Syslog Reference Identifier Parameters on TOE. • Verify that the failure is audited and the reason for failure is reflected in the audit log.
Expected Test Results	When an attempt to modify TOE Certificate Trust Store Parameter using an unprivileged user, it should result in failure as it is not the Security Administrator. Audit log confirms the user to not have prior authentication as security administrator.
Pass/Fail with Explanation	Pass. Users without administrator privilege were not able to modify parameters/services on the TOE. This meets testing requirements.

7.2.33 FMT_MOF.1/FUNCTIONS (1)TEST #2

Item	Data
Test Assurance Activity	Test 2 (if **'transmission of audit data to external IT entity'** is selected from the second selection together with **'modify the behaviour of'** in the first selection): The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity with prior authentication as Security Administrator. The effects of the modifications should be confirmed. The evaluator does not have to test all possible values of the security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity but at least one allowed value per parameter.
Audit Log Requirement	[PP] FMT_MOF.1/Functions: There are no audit logs for FMT_MOF.1/Functions. [PP] FMT_SMF.1: Audit log is required when the behaviour of the transmission of audit data to an external IT entity is modified.
Test Steps	• Login to the TOE as a user with administrator privileges. • Attempt to modify Syslog Reference Identifier Configuration Parameter on TOE. • Verify that the said parameter is successfully modified and audited in logs.

Expected Test Results	When an administrator tries to modify the audit data on the TOE, it should be successful. The command should be executed as the user has administrator privileges.
Pass/Fail with Explanation	Pass. Users with administrator privileges were able to modify services on TOE. This meets the testing requirements.

7.2.34 FMT_MOF.1/FUNCTIONS (2) TEST #1

Item	Data
Test Assurance Activity	Test 1 (if **'handling of audit data'** is selected from the second selection together with **'modify the behaviour of'** in the first selection): The evaluator shall try to modify all security related parameters for configuration of the handling of audit data without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters 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 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 term 'handling of audit data' refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace.
Pass/Fail with Explanation	NA. Handling of audit data is not selected. Hence, this test is not applicable for this TOE.

7.2.35 FMT_MOF.1/FUNCTIONS (2) TEST #2

Item	Data
Test Assurance Activity	Test 2 (if **'handling of audit data'** is selected from the second selection together with **'modify the behaviour of'** in the first selection): The evaluator shall try to modify all security related parameters for configuration of the handling of audit data with prior authentication as Security Administrator. The effects of the modifications should be confirmed. The term

	'handling of audit data' refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace. The evaluator does not necessarily have to test all possible values of the security related parameters for configuration of the handling of audit data but at least one allowed value per parameter.
Pass/Fail with Explanation	NA. Handling of audit data is not selected. Hence, this test is not applicable for this TOE.

7.2.36 FMT_MOF.1/FUNCTIONS (3) TEST #1

Item	Data
Test Assurance Activity	(if '**audit functionality when Local Audit Storage Space is full**' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify the behaviour when Local Audit Storage Space is full without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). This attempt 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 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.
Pass/Fail with Explanation	NA. Behavior of audit functionality in the TOE cannot be modified when local audit storage is full. Hence, this test is not applicable for this TOE.

7.2.37 FMT_MOF.1/FUNCTIONS (3) TEST #2

Item	Data
Test Assurance Activity	(if '**audit functionality when Local Audit Storage Space is full**' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify the behaviour when Local Audit Storage Space is full with prior authentication

	as Security Administrator. This attempt should be successful. The effect of the change shall be verified. The evaluator does not necessarily have to test all possible values for the behaviour when Local Audit Storage Space is full but at least one change between allowed values for the behaviour
Pass/Fail with Explanation	NA. Behavior of audit functionality in the TOE cannot be modified when local audit storage is full. Hence, this test is not applicable for this TOE.

7.2.38 FMT_MOF.1/FUNCTIONS TEST #3

Item	Data
Test Assurance Activity	(if in the first selection **'determine the behaviour of'** has been chosen together with for any of the options in the second selection): The evaluator shall try to determine the behaviour of all options chosen from the second selection without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). This can be done in one test or in separate tests. The attempt(s) to determine the behaviour of the selected functions without administrator authentication shall 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 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.
Pass/Fail with Explanation	NA. 'Determine the behavior of' option is not selected in the ST. Hence, this test is not applicable to the TOE.

7.2.39 FMT_MOF.1/FUNCTIONS TEST #4

Item	Data
Test Assurance Activity	(if in the first selection **'determine the behaviour of'** has been chosen together with for any of the options in the second selection): The evaluator shall try to determine the behaviour of all options chosen from the second selection with prior authentication as Security Administrator. This can be done in one test or in separate tests. The attempt(s) to determine the behaviour of the selected functions with Security Administrator authentication shall be successful.
Pass/Fail with Explanation	NA. 'Determine the behavior of' option is not selected in the ST. Hence, this test is not applicable to the TOE.

7.2.40 FMT_MTD.1/CRYPTOKEYS TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FMT_MOF.1/Functions: There are no audit logs for FMT_MOF.1/CryptoKeys. [PP] FAU_GEN.1: When crypto keys are imported/deleted/replaced/generated, an audit log is required. The audit log should uniquely identify the key.
Test Steps	Crypto Key Generation using CSR: • Login into the TOE with an unprivileged user. • Verify the generating of CSR fails for unprivileged users. • Check for any audit logs for this event.
Expected Test Results	Non-administrative user should not make modifications to cryptographic keys (modify, delete, generate/import) on TOE.

Pass/Fail with Explanation	Pass. Non-Administrative users cannot download CSR OR Upload CA on trusted store of TOE. This meets testing requirements.

7.2.41 FMT_MTD.1/CRYPTOKEYS TEST #2

Item	Data
Test Assurance Activity	The evaluator shall try to perform at least one of the related actions with prior authentication as Security Administrator. This attempt should be successful.
Audit Log Requirement	[PP] FMT_MOF.1/Functions: There are no audit logs for FMT_MOF.1/CryptoKeys. [PP] FAU_GEN.1: When crypto keys are imported/deleted/replaced/generated, an audit log is required. The audit log should uniquely identify the key.
Test Steps	• Login into the TOE with a privileged user. • Verify the generating of CSR for privileged users. • Verify that the successful of the generating CSR is audited in the audit log.
Expected Test Results	Attempts to perform related actions with prior authentication should Pass.
Pass/Fail with Explanation	Pass. The TOE allows the admin user to upload certificates and successfully log these actions. This meets testing requirements.

7.2.42 FMT_SMF.1 TEST #1

Item	Data
Test Assurance Activity	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.
Notes	**The NiT has issued a technical decision (TD0571) for Guidance on how to handle FIA_AFL.1.** 1. FIA_UAU_EXT.2.1 applies solely to how the administrator logs in at the local console. Passwords were made selection based to allow TOEs that have difficulty

	determining whether an incoming connection is local or remote to provide a mechanism to prevent administrative lockout. As FIA_AFL.1 is a mandatory SFR, it is expected that the TOE provides at least one remote password-based authentication mechanism using credentials managed by the TOE. FIA_PMG_EXT.1, FIA_AFL.1, and FMT_SMF.1 are all mandatory SFRs that the TOE will need to meet.
Audit Log Requirement	[PP] FMT_SMF.1: All management activities shall result in audit logs
Test Steps	The TSF shall be capable of performing the following management functions: • Ability to administer the TOE locally and remotely • Ability to configure the access banner • Ability to configure the session inactivity time before session termination or locking • Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates • Ability to configure the authentication failure parameters for FIA_AFL.1 • Ability to configure audit behavior • Ability to manage the cryptographic keys • Ability to re-enable an Administrator account • Ability to set the time which is used for timestamps • Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors. • Ability to import X.509v3 certificates to the TOE's trust store.
Expected Test Results	All management functions identified in section 2.4.4 have been tested throughout the evaluation. Thus, this requirement has been met.
Pass/Fail with Explanation	Pass. All management functions identified in section 2.4.4 have been tested throughout the evaluation. This meets requirements.

7.2.43 FMT_SMR.2 TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FMT_SMF.1: All management activities shall result in an audit log
Test Steps	As there are two interfaces where these can be tested (over the GUI/Console) and all test cases are tested that way. The evaluator has met this requirement through execution of the entirety of this test report for the TOE interfaces.
Expected Test Results	As there are two interfaces where these can be tested (over the GUI/Console) and all test cases are tested that way. The evaluator has met this requirement through execution of the entirety of this test report for the TOE interfaces.
Pass/Fail with Explanation	Pass. This test requirement has been performed in conjunction with other tests.

7.2.44 FTA_SSL.3 TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FTA_SSL.3: Termination of a remote session by the session locking mechanism. [PP] FMT_SMF.1: When configuring the session inactivity time before session termination or locking
Test Steps	GUI: • Configure a remote GUI out period of 2 minutes on administrative sessions. • Connect to the TOE from the remote GUI. • Let the remote GUI connection be idle for 2 minutes. • Verify that the session is terminated. • Verify that the termination of session is audited.

	• Configure a remote GUI out period of 4 minutes on administrative sessions. • Connect to the TOE from the remote GUI. • Let the remote GUI connection be idle for 4 minutes. • Verify that the session is terminated. • Verify that the termination of session is audited.
Expected Test Results	The TOE should terminate idle remote sessions after the specified time. Time of audit log indicating 'Automatic logout due to Keyboard inactivity' shows auto logout of session after TOE is idle for specified period of time.
Pass/Fail with Explanation	Pass. Evaluator observed that session is being timeout, where no activity performed during configured session timeout value on TOE. This meets requirements.

7.2.45 FTA_SSL.4 TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FTA_SSL.4: Termination of an interactive session. [PP] FAU_GEN.1: Administrator login and logout
Test Steps	• Log onto the TOE through a directly connected interface. • Using the instructions provided by the user guide, log off the TOE. • Verify that the termination of session is audited.
Expected Test Results	The user is getting logged in via directly connected interface on TOE and information provided by user guide TOE terminates the session post user logged out.
Pass/Fail with Explanation	Pass. The evaluator initiated an interactive local session with the TOE by following the guidance documentation, also logged out the session and observed that the session has been terminated. This meets testing requirements.

7.2.46 FTA_SSL.4 TEST #2

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FTA_SSL.4: Termination of an interactive session. [PP] FAU_GEN.1: Administrator login and logout
Test Steps	GUI: • Log onto the TOE through a remote GUI interface. • Using the instructions provided by the user guide log off. • Verify that the termination of session is audited.
Expected Test Results	The TOE should allow users to terminate the remote sessions. Audit logs show the successful login and logout of user from TOE.
Pass/Fail with Explanation	Pass. The TOE allows users to terminate remote administrative sessions. This meets the testing requirements.

7.2.47 FTA_SSL_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	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 reauthentication is needed when trying to unlock the session.
Audit Log Requirement	[PP] FTA_SSL.3: Termination of a local session by the session locking mechanism. [PP] FMT_SMF.1: When configuring the session inactivity time before session termination or locking.

Test Steps	Console: - • Configure a local time out period of 2 minutes on administrative sessions. • Connect to the TOE from the local connection. • Let the local connection remain idle for 2 minutes and check that it terminates after 2 minutes. • Verify that the termination of session is audited. • Verify that re-authentication is needed to unlock the session. • Configure a local time out period of 4 minutes on administrative sessions. • Connect to the TOE from the local connection. • Let the local connection remain idle for 4 minutes and check that it terminates after 4 minutes. • Verify that the termination of session is audited. • Verify that Re-authentication is needed to unlock the session.
Expected Test Results	The TOE should terminate idle local sessions after the specified time. Time of audit log indicating 'Automatic logout due to Keyboard inactivity' shows auto logout of session after TOE is idle for specified period of time.
Pass/Fail with Explanation	Pass. For each period configured, the evaluator has established local interactive session with the TOE and then the evaluator has observed that the session was terminated after the configured time period. This meets testing requirements.

7.2.48 FTA_TAB.1 TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FTA_TAB.1: There are no audit logs for FTA_TAB.1 [PP] FMT_SMF.1: When the access banner is configured.
Test Steps	GUI: • Login to the TOE via GUI and configure the banner. • Verify that the configuring of banner messages is audited.

	• Logoff and login again and verify that banner is being displayed. Console: • Login to the TOE using console & verify that the banner is being displayed while login.
Expected Test Results	When any user accesses the TOE through the console or GUI, the configured banner should be displayed prior to authenticating the TOE.
Pass/Fail with Explanation	Pass. Banner is displayed while accessing TOE using all the access methods specified. This meets testing requirements.

7.2.49 FTP_TRP.1/ADMIN TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FTP_TRP.1/Admin: • Initiation of the trusted path • Termination of the trusted path • Failure of the trusted path [PP] FMT_SMF.1: When TSF Parameters are managed.
Test Steps	GUI: • Start an administrative session with the device over HTTPS. • Capture the packets between the remote workstation and the TOE and verify that the connection is successful. • Verify that the initiation and termination of session is audited.
Expected Test Results	Successful communication between TOE and remote administrator via HTTPS. Application Data packets in HTTPS connection and Encrypted Packets connection in packet capture confirms successful connection.

Pass/Fail with Explanation	Pass. Remote administrative access to the TOE is over secured channels. This meets the testing requirements.

7.2.50 FTP_TRP.1/ADMIN TEST #2

Item	Data
Test Assurance Activity	The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext.
Audit Log Requirement	[PP] FTP_TRP.1/Admin: • Initiation of the trusted path • Termination of the trusted path • Failure of the trusted path [PP] FMT_SMF.1: When TSF Parameters are managed.
Test Steps	This test is performed in conjunction with FTP_TRP.1/Admin Test #1 test.
Expected Test Results	This test is performed in conjunction with FTP_TRP.1/Admin Test #1 test.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FTP_TRP.1/Admin Test #1 test. Remote administrative access to the TOE is over secured channels and the data was not sent in plaintext. This meets the testing requirements.

7.3 TLSC

7.3.1 FCS_TLSC_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	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 in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no audit logs required for this test to capture because no failures are expected. [PP] FMT_SMF.1: If the characteristics of the remote log settings are changed, then these changes need to be audited.
Test Steps	• Attempt a connection from TOE To Server with TLS_RSA_WITH_AES_128_CBC_SHA cipher suite. • Verify with packet capture that the connection is successfully established, and that application data is flowing. • Attempt a connection from TOE To Server with TLS_RSA_WITH_AES_256_CBC_SHA cipher suite. • Verify with packet capture that the connection is successfully established, and that application data is flowing. • Attempt a connection from TOE To Server with TLS_RSA_WITH_AES_128_CBC_SHA256 cipher suite. • Verify with packet capture that the connection is successfully established, and that application data is flowing. • Attempt a connection from TOE To Server with TLS_RSA_WITH_AES_256_CBC_SHA256 cipher suite. • Verify with packet capture that the connection is successfully established, and that application data is flowing. • Attempt a connection from TOE To Server with TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 cipher suite. • Verify with packet capture that the connection is successfully established, and that application data is flowing. • Attempt a connection from TOE To Server with TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 cipher suite. • Verify with packet capture that the connection is successfully established, and that application data is flowing.

Expected Test Results	TOE should successfully establish connections with a TLS Server with each of below cipher suites: ○ TLS_RSA_WITH_AES_128_CBC_SHA ○ TLS_RSA_WITH_AES_256_CBC_SHA ○ TLS_RSA_WITH_AES_128_CBC_SHA256 ○ TLS_RSA_WITH_AES_256_CBC_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
Pass/Fail with Explanation	Pass. TOE was successfully able to establish TLS sessions with a TLS Server with specified ciphersuites. This meets testing requirements.

7.3.2 FCS_TLSC_EXT.1.1 TEST #2

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure.
Test Steps	• Create a server certificate with the Server Authentication EKU. • Attempt a connection from the TOE to a TLS server using a valid certificate that contains the Server Authentication EKU. • Verify with packet capture that the connection is successfully established, and that application data is flowing. • Create a server certificate that lacks the Server Authentication EKU. • Attempt a connection from the TOE to a TLS server using an invalid certificate missing the Server Authentication EKU. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	• The TOE should establish a connection with Server using certificate contains “server authentication purpose” in the Extended Key usage field of certificate.

	• The TOE should reject the connection with Server due to lack of “server authentication purpose” in the Extended Key usage field of certificate.
Pass/Fail with Explanation	Pass. The TOE accepts the connection with a server with a Server Authentication extended keyusage field and the TOE rejects the connection with a server without a Server Authentication extended keyusage field. This meets the testing requirements.

7.3.3 FCS_TLSC_EXT.1.1 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send a 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.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure.
Test Steps	• Use Acumen TLSC tool to attempt a TLS connection to the TOE with a certificate that doesn't match the server selected cipher suite. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The TOE should deny the TLS connection with server, if the certificate sent by the server does not match the cipher suite.
Pass/Fail with Explanation	Pass. The TOE denied a connection to a server using a certificate that doesn’t match the ciphersuite. This meets the testing requirements.

7.3.4 FCS_TLSC_EXT.1.1 TEST #4A

Item	Data

Test Assurance Activity	The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure.
Test Steps	• Attempt a TLS connection with TOE using Acumen-TLSC tool with TLS_NULL_WITH_NULL NULL cipher suite and wait for the connection, the connection should fail. • Verify that the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The TOE should not make the TLS connection because the cipher suite present in server certificate was TLS_NULL_WITH_NULL_NULL.
Pass/Fail with Explanation	Pass. A TLS connection refused by TOE as Server sent certificate to the TOE with TLS_NULL_WITH_NULL NULL cipher suite. This meets testing requirements.

7.3.5 FCS_TLSC_EXT.1.1 TEST #4B

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure.
Test Steps	• Attempt a connection from the TOE to a remote TLS server using Acumen-TLSC tool that would allow the server's cipher suite to be modified to unsupported cipher and shows that the connection fails. • Verify that the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	TOE should reject the connection when unsupported cipher suite is sent in the server hello message.

Pass/Fail with Explanation	Pass. The console output shows the Acumen-TLS tool modifying the servers selected cipher suite in the Server Hello message to one that is not present in the Client Hello. The TOE rejects the connection by sending a Fatal Alert. This meets the test requirements.

7.3.6 FCS_TLSC_EXT.1.1 TEST #4C

Item	Data
Test Assurance Activity	[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.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure.
Test Steps	• Attempt a TLS connection to the TOE from server with Acumen-TLSC tool using non-supported curve/group. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The Acumen-TLSC tool is used to establish a TLS server connection with the TOE using an unsupported curve and the TOE should drop the connection. The packet capture shows the supported curves and then the unsupported curve is used to establish the connection. The logs describe effectively describe that the connection was dropped due to an unknown curve group.
Pass/Fail with Explanation	Pass. When configured the server to perform an ECDHE key exchange in the TLS connection using a non-supported curve, TOE rejects the connection. This meets the requirements.

7.3.7 FCS_TLSC_EXT.1.1 TEST #5A

Item	Data

Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure.
Test Steps	• Using Acumen-TLSC tool, attempt a connection to a remote TLS server using a non-supported TLS version and verify that the TOE rejects the connection. • Verify that the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The Acumen-TLSC tool is used to establish a TLS server connection with the TOE using an unsupported TLS version. The TOE rejects the connection when it detects that the TLS version used is unsupported. The packet capture shows the tls version used to establish the connection and then dropping the connection. The logs confirm that the connection has been terminated due to incorrect version number.
Pass/Fail with Explanation	Pass. The connection was rejected due to unsupported TLS version. This meets the test requirements.

7.3.8 FCS_TLSC_EXT.1.1 TEST #5B

Item	Data
Test Assurance Activity	[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.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure. [PP] FMT_SMF.1: If the characteristics of the remote log settings are changed, then these changes need to be audited.
Test Steps	• Attempt a connection from the TOE to a remote TLS server using Acumen-TLSC tool that would allow the server's signature block to be modified and shows that the connection fails.

	• Verify that the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The Acumen-TLSC tool is used to establish a TLS server connection with the TOE. The tool is used to change the signature in the Server's Key exchange message for DHE or ECDH cipher. The TOE rejects the connection when it detects that the signature is modified. The capture should show that the connection has been dropped due to a decrypt error and the logs confirm that the connection has been disconnected.
Pass/Fail with Explanation	Pass. The TOE rejects due to the modified signature block in the Server Key Exchange message. This meets the test requirement.

7.3.9 FCS_TLSC_EXT.1.1 TEST #6A

Item	Data
Test Assurance Activity	Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application data flows.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure.
Test Steps	• Attempt a connection from the TOE to a remote TLS server using Acumen-TLSC tool that would allow modify a byte in the server finish handshake message and shows that the connection fails. • Verify that the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The Acumen-TLSC tool is used to establish a TLS server connection with the TOE. The tool is used to modify a byte in the Server Finished handshake message. When the TOE detects that the message has been modified, it rejects the connection. The packet should show that the connection has been dropped after a modified Server finished message is sent. The logs confirm that the connection has been terminated.
Pass/Fail with Explanation	Pass. The connection is rejected when a corrupted Server Finished message is received by TOE. This meets the test requirements.

7.3.10 FCS_TLSC_EXT.1.1 TEST #6B

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure.
Test Steps	• Attempt a TLS connection to Server using Acumen-TLSC that would allow sending a garbled message from the server before the server issues the Change Cipher Spec message and shows that the TOE rejects the connection. • Verify that the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The Acumen-TLSC tool is used to establish a TLS server connection. The tool is used to send a garbled message after the server has issued Change Cipher Spec message. When the TOE receives the garbled message, it drops the connection by sending an 'Encrypted Alert'. The packet capture should show that the connection has been concluded and the logs should confirm that the connection has been disconnected.
Pass/Fail with Explanation	Pass. The TOE rejected TLS connection with server after receiving garbled data. This meets the test requirements.

7.3.11 FCS_TLSC_EXT.1.1 TEST #6C

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure.

Test Steps	• Attempt a connection from the TOE to a remote TLS server using Acumen-TLSC tool that would allow the modification in the Server nonce of server hello handshake message and shows connection should be rejected by the TOE. • Verify that the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The 'Acumen-TLSC' tool is used to establish a TLS server connection with the TOE. The tool modifies server nonce byte in the Server Hello Handshake message, and this results in the TOE rejecting the connection. The packet capture depicts that the connection is terminated when the TOE realizes that the Server Hello Handshake has been modified. The logs confirm that the connection has been dropped due to a decryption error.
Pass/Fail with Explanation	Pass. The connection was rejected by the TOE due to modified nonce of server hello handshake message. This meets the test requirements.

7.3.12 FCS_TLSC_EXT.1.2 TEST #1

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** 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.

Notes	*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.*
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure. [PP] FMT_SMF.1: The configuration of the reference identifier on the TOE should be audited.
Test Steps	**FQDN: Invalid CN, No SAN** • Configure the correct FQDN reference identifier in the TOE. • Verify that there is an audit log reflecting the change in the reference identifier. • Create a server certificate with **invalid CN and no SAN.** • Connect to the TLS Server using the mismatched CN and shows that it fails. • Verify that the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	When the CN configured on server certificated doesn't match the reference identifier configured on the TOE, the TOE should reject the connection. It issues an alert of 'internal error'. The packet capture should confirm that the connection is terminated by the TOE and the logs should validate that the connection has been concluded.
Pass/Fail with Explanation	Pass.

	The TOE rejects the TLS connection with server for certificate with an Invalid CN and No SAN. This meets the testing requirements.

7.3.13 FCS_TLSC_EXT.1.2 TEST #2

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** 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.
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *d) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *e) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable* *or* *f) 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.*

Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure. [PP] FMT_SMF.1: The configuration of the reference identifier on the TOE should be audited.
Test Steps	**FQDN: Valid CN, Invalid SAN** • Configure the correct FQDN reference identifier in the TOE. • Create a server certificate with **valid FQDN CN but invalid FQDN SAN.** • Attempt a connection to the TLS server and shows that it fails. • Verify that the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	When a server certificate contains a CN that matches the reference identifier configured on TOE, but the SAN configured on the server certificate doesn't match the reference identifier, then the TOE should reject the connection. It should issue an alert of 'certificate unknown'. The packet capture shows that connection rejected, and the logs confirm that the connection is terminated when there is a mismatch between reference identifier and SAN.
Pass/Fail with Explanation	Pass. The TOE rejects the TLS connection with server for certificate with a valid CN and invalid SAN. This meets the testing requirements.

7.3.14 FCS_TLSC_EXT.1.2 TEST #3

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** 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.

Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *g) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *h) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable* *or* *i) 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.*
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no audit logs required for this test to capture because no failures are expected. [PP] FMT_SMF.1: The configuration of the reference identifier on the TOE should be audited.
Test Steps	**FQDN: Valid CN, No SAN** • Configure the correct FQDN reference identifier in the TOE. • Create a server certificate with a **valid FQDN CN and no SAN**. • Connect to the TLS Server using the certificate with the valid CN and show that the connection is successful. • Verify with packet capture that the connection is established successfully, and that application data is flowing.
Expected Test Results	The TOE establishes a successful TLS server connection when there is no SAN but correct CN is configured in the server certificate which matches the reference identifier configured on TOE. The packet capture confirms the successful connection.
Pass/Fail with Explanation	Pass. The TOE successfully accepts the TLS connection when the certificate with a Good CN and No SAN is presented. This meets the testing requirements.

7.3.15 FCS_TLSC_EXT.1.2 TEST #4

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** 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).
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *j) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *k) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable* *or* *l) 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.*
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no audit logs required for this test to capture because no failures are expected. [PP] FMT_SMF.1: The configuration of the reference identifier on the TOE should be audited.
Test Steps	**FQDN: Invalid CN, Valid SAN**

	• Configure the correct FQDN reference identifier in the TOE. • Create a server certificate with an **invalid FQDN CN and a valid FQDN SAN**. • Connect to the TLS Server using the certificate with the valid CN and show that the connection is successful. • Verify with packet capture that the connection is established successfully, and that application data is flowing.
Expected Test Results	The TOE establishes successful TLS server connection when the server certificate that has an invalid CN but has a valid SAN which matches the reference identifier configured on TOE. The packet capture confirms the same and shows that a successful connection has been established.
Pass/Fail with Explanation	Pass. The TOE successfully accepts the TLS connection when the certificate with a bad CN and valid SAN is presented. This meets the testing requirements.

7.3.16 FCS_TLSC_EXT.1.2 TEST #5 (1)

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** Test 5: 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, URI-ID): 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.
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *m) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *n) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable* *or*

	o) *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.*
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure. [PP] FMT_SMF.1: If the characteristics of the remote log settings are changed, then these changes need to be audited.
Test Steps	**CN-ID with DNS:** • Configure a correct DNS reference identifier with a SINGLE left-most label (**rsyslog.acumen.com)** in the TOE. • Create a server certificate containing a **DNS CN which has a wildcard that is not in the left-most label (and no SAN)**. • Show that the connection attempt fails. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection attempt Is unsuccessful. **SAN-ID with DNS:** • Configure the DNS correct reference identifier with a single left-most label (**rsyslog.acumen.com**) in the TOE. • Create a server certificate containing a **DNS SAN which has a wildcard that is not in the left-most label**. • Show that the connection attempt fails. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The TOE should reject the TLS server connection as the wildcard does not match with the reference identifier configured on TOE. When the TOE rejects the connection, it issues an

	alert of 'certificate unknown'. The packet capture confirms the same and logs depict that the connection was dropped as the TOE wasn't able to verify the certificate.
Pass/Fail with Explanation	Pass. TOE rejects the connection when the reference identifier does not match the presented wildcard which is not in the leftmost label. This meets the testing requirements.

7.3.17 FCS_TLSC_EXT.1.2 TEST #5 (2)(A)

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** Test 5: 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, URI-ID): 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. (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.)
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *p) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *q) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable* *or*

	r) *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.*
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no audit logs required for this test to capture because no failures are expected. [PP] FMT_SMF.1: The configuration of the reference identifier on the TOE should be audited.
Test Steps	**CN-ID with DNS:** • Configure a correct DNS reference identifier with a SINGLE left-most label (**rsyslog.acumen.com)** in the TOE. • Create a server certificate containing a **DNS CN which has a wildcard in the left-most label (and no SAN)**. • Attempt to connect to the TOE and show that the connection is successful. • Verify with packet capture that the connection is successful, and that application data is flowing. **SAN-ID with DNS:** • Configure a correct DNS reference identifier with a SINGLE left-most label (**rsyslog.acumen.com)** in the TOE. • Create a server certificate containing a **DNS SAN which has a wildcard in the left-most label**. • Attempt to connect to the TOE and show that the connection is successful. • Verify with packet capture that the connection is successful, and that application data is flowing.
Expected Test Results	The TOE establishes a successful TLS Server connection as the reference identifier matches with the wildcard that has been configured in the server certificate. The packet capture helps to confirm that the reference identifier matches with the wildcard configured in the server certificate.

Pass/Fail with Explanation	Pass. TOE accepts a connection when a server certificate that has a wildcard in the left-most label is presented to the TOE while a reference identifier with single left-most label is configured in the TOE. This meets the testing requirements.

7.3.18 FCS_TLSC_EXT.1.2 TEST #5 (2)(B)

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** Test 5: 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, URI-ID): 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 without a left-most label as in the certificate (e.g. 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.)
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *s) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *t) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable* *or* *u) 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.*
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure. [PP] FMT_SMF.1: The configuration of the reference identifier on the TOE should be audited.
Test Steps	**CN-ID with DNS:** • Configure a correct DNS reference identifier WITHOUT a left-most label (**acumen.com**) in the TOE. • Create a server certificate containing a **DNS CN which has a wildcard in the left-most label (and no SAN)**. • Attempt to connect to the TOE and show that the connection fails. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful. **SAN-ID with DNS:** • Configure a correct DNS reference identifier WITHOUT a left-most label (**acumen.com**) in the TOE. • Create a server certificate containing a **DNS SAN which has a wildcard in the left-most label**. • Attempt to connect to the TOE and show that the connection fails. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	When the reference identifier configured on the TOE doesn't match the wildcard configured on the certificate, the TOE should drop the TLS server connection by issuing an alert of 'internal error'. The packet shows that connection could not be established, and the logs depict that the connection has been rejected.
Pass/Fail with Explanation	Pass.

	When a server certificate with a wildcard in the left-most label is presented to the TOE which has a reference identifier without a left-most label configured, the connection gets rejected. This meets the testing requirements.

7.3.19 FCS_TLSC_EXT.1.2 TEST #5 (2)(C)

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT.** Test 5: 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, URI-ID): 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 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.)
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *v) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *w) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable* *or* *x) 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.*
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure. [PP] FMT_SMF.1: The configuration of the reference identifier on the TOE should be audited.
Test Steps	**CN-ID with DNS:** • Configure a correct DNS reference identifier with **TWO** left-most labels (**acucert.rsyslog.acumen.com**) in the TOE. • Create a server certificate containing a **DNS CN which has a wildcard in the left-most label (and no SAN)**. • Attempt to connect to the TOE and show that the connection fails. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful. **SAN-ID with DNS:** • Configure a correct DNS reference identifier with **TWO** left-most labels (**acucert.rsyslog.acumen.com**) in the TOE. • Create a server certificate containing a **DNS SAN which has a wildcard in the left-most label**. • Attempt to connect to the TOE and show that the connection fails. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	When the reference identifier configured on TOE don't match the wildcards used, the TOE should issue an alert of 'internal error ' and fail to establish a TLS server connection. The packet capture should show that the connection is dropped, and the logs confirm that the connection has been terminated as the presented certificate could not be verified.
Pass/Fail with Explanation	Pass. When configuring the reference identifier with two left-most labels on TOE, the connections are rejected by TOE. This meets the testing requirements.

7.3.20 FCS_TLSC_EXT.1.2 TEST #6 [TD0790]

Item	Data
Test Assurance Activity	This test is applicable if **TLS-based communications with RFC 6125 is selected for FTP_ITC.1, FTP_TRP, or FPT_ITT**. If IP address identifiers are supported in the SAN or CN, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with a wildcard asterisk (*) (e.g. CN=*.168.0.1 when connecting to 192.168.0.1, CN=2001:0DB8:0000:0000:0008:0800:200C:* when connecting to 2001:0DB8:0000:0000:0008:0800:200C:417A). The certificate shall not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported IP address version (e.g. IPv4, IPv6). Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 6. **TD0790 has been applied.**
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *y) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *z) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable* *or* *aa) 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.*
Pass/Fail with Explanation	NA. The ST does not claim IP address identifiers in the SAN or CN. Hence, this test is not applicable to the TOE.

7.3.21 FCS_TLSC_EXT.1.2 TEST #7A

Item	Data
Test Assurance Activity	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): 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.
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *bb) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *cc) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable* *or* *dd) 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.*
Pass/Fail with Explanation	NA. This is not a distributed TOE. Hence, this test is not applicable to the TOE.

7.3.22 FCS_TLSC_EXT.1.2 TEST #7B

Item	Data
Test Assurance Activity	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): 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-at-serialNumber=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.
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *ee) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or*

	ff) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable *or* *gg) 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.*
Pass/Fail with Explanation	NA. This is not a distributed TOE. Hence, this test is not applicable to the TOE.

7.3.23 FCS_TLSC_EXT.1.2 TEST #7C

Item	Data
Test Assurance Activity	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): 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.
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *hh) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or*

	ii) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable *or* *jj) 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.*
Pass/Fail with Explanation	NA. This is not a distributed TOE. Hence, this test is not applicable to the TOE.

7.3.24 FCS_TLSC_EXT.1.2 TEST #7D

Item	Data
Test Assurance Activity	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): 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.)
Notes	*Note that the following tests are marked conditional and are applicable under the following conditions:* *kk) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or*

	ll) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable *or* *mm) 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.*
Pass/Fail with Explanation	NA. This is not a distributed TOE. Hence, this test is not applicable to the TOE.

7.3.25 FCS_TLSC_EXT.1.3 TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no audit logs required for this test to capture because no failures are expected. [PP] FMT_SMF.1: The configuration of the reference identifier of the TLS server on the TOE should be audited. Any changes that modify the behaviour of the transmission of audit data to the remote syslog server should be audited. Adding, Modifying, or Deleting, X509 certificates in the certificate trust store of the TOE should be audited.

Test Steps	• Configure TOE to connect to the Syslog Server. • Verify that the changes to the syslog server settings on the TOE are audited. • Create a full chain of certificates to connect to the TOE. • Upload a complete certificate validation chain to the TOE. • Verify that the import of the certificates results in an audit log which identifies which certificates got imported. • With the whole chain present, attempt the connection from the TOE to the TLS server and show the connection is successful. • Verify with psacket capture that the connection is successful, and that application data is flowing.
Expected Test Results	While making a connection between TOE and TLS server we should see complete certificate chain present and required connection should established between TOE and TLS server.
Pass/Fail with Explanation	Pass. When a complete certificate trust chain is present, the TOE can make a successful connection. This meets the test requirements.

7.3.26 FCS_TLSC_EXT.1.3 TEST #2

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSC_EXT.1: There are expected failures in this test that should result in an audit log which reflects the reason for failure. [PP] FMT_SMF.1: The configuration of the reference identifier on the TOE should be audited. Adding, Modifying, or Deleting, X509 certificates in the certificate trust store of the TOE should be audited.

Test Steps	**Failed matching reference Identifier:** • The requirements of this test case are exercised in in FCS_TLSC_EXT.1.2 Test #1 and Test #2. **Failed Certificate Path:** • Remove the good Intermediate CA certificate and add Imposter Intermediate CA certificate in chain on the TOE. • Verify that the deletion of the Intermediate CA certificate is recorded with an audit log which identifies which certificate was deleted. • Attempt the connection from the TOE to the TLS server and show the connection is unsuccessful. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful. **Expired Certificate:** • Verify that the date and time on the TOE is current. • Create an expired Intermediate CA certificate. • Attempt to import the newly created expired Intermediate CA certificate into the TOE's trust store and show that import is successful. • Verify that the successful Intermediate CA certificate import attempt is audited. • Attempt to establish a TLS connection with valid Server Certificates while the expired Intermediate CA certificate is in the TOE and show the connection is not successful. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful. • Replace the expired Intermediate CA certificate with a valid Intermediate CA certificate. • Using a valid server certificate, make a TLS connection with the TOE and show that the connection is successful. • Create a server certificate which is already expired. • Attempt to establish a TLS connection with the TOE using the expired server certificate and show the connection is not successful. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful. • Remove the good Intermediate CA certificate and upload Imposter Intermediate CA certificate and good Root CA on the TOE. • Attempt to establish a TLS connection with the TOE using a valid server certificate AND an expired Intermediate CA certificate and show that the connection is not successful. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log.

	• Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	This test should meet requirements (Failed Certificate Path & Expired Certificate). In Failed Certificate Path we should not see Signing Certificate and in Expired Certificate we should see server certificate expired and, in both cases, TOE rejecting connection with to server.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when Invalid certificates are presented. This meets the test requirements.

7.3.27 FCS_TLSC_EXT.1.3 TEST #3

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	NA. The TOE does not provide for, nor claim, any administrator-defined override mechanism for validating the certificate path for claimed TLS channels. Therefore, this test is applicable in the context of FCS_TLSC_EXT.1. For all other uses of X.509 certificates, use of an override mechanism is not permitted. For more information, please refer to FCS_TLSC_EXT.1.3 in the Protection Profile.

7.3.28 FCS_TLSC_EXT.1.4 TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no audit logs required for this test to capture because no failures are expected. [PP] FMT_SMF.1: If the characteristics of the remote log settings are changed, then these changes need to be audited.
Test Steps	• Initiate the connection from the TOE to the TLS Server using the curve secp256r1 and show the connection is successful. • Verify with packet capture that the connection successful and application data is flowing. • Initiate the connection from the TOE to the TLS Server using the curve secp384r1 and show the connection is successful. • Verify with packet capture that the connection successful and application data is flowing. • Initiate the connection from the TOE to the TLS Server using the curve secp521r1 and show the connection is successful. • Verify with packet capture that the connection successful and application data is flowing.
Expected Test Results	TOE should accept connections with supported EC (secp256r1, secp384r1 & secp521r1) from TLS server.
Pass/Fail with Explanation	Pass. The TOE accepted a connection when supported curves were introduced. This meets the test requirements.

7.4 TLSS

7.4.1 FCS_TLSS_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	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).
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no failures in this test to capture because no failures are expected.
Test Steps	• Use openssl Tool to establish a connection with the TOE over TLS_RSA_WITH_AES_128_CBC_SHA cipher suite. • Verify with packet capture that the connection is successful, and that the application data is flowing. • Use openssl Tool to establish a connection with the TOE over TLS_RSA_WITH_AES_256_CBC_SHA cipher suite. • Verify with packet capture that the connection is successful, and that the application data is flowing. • Use openssl Tool to establish a connection with the TOE over TLS_RSA_WITH_AES_128_CBC_SHA256 cipher suite. • Verify with packet capture that the connection is successful, and that the application data is flowing. • Use openssl Tool to establish a connection with the TOE over TLS_RSA_WITH_AES_256_CBC_ SHA256 cipher suite. • Verify with packet capture that the connection is successful, and that the application data is flowing. • Use openssl Tool to establish a connection with the TOE over TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 cipher suite. • Verify with packet capture that the connection is successful, and that the application data is flowing. • Use openssl Tool to establish a connection with the TOE over TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 cipher suite. • Verify with packet capture that the connection is successful, and that the application data is flowing.
Expected Test Results	• OpenSSL Client should successfully be able to establish a connection with Server (TOE) for below cipher suites: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256

	• TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
Pass/Fail with Explanation	Pass. The TOE was able to make each connection via the supported cipher suites. This meets the test requirements.

7.4.2 FCS_TLSS_EXT.1.1 TEST #2

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: Failure to establish a TLS Session shall be audited. Audit log shall reflect the reason for failure.
Test Steps	Use the "openssl & acumen-tlss" tool to initiate a connection to the TOE and verify the connection fails with the non-supported cipher suites. • Attempt to establish a TLS connection to the TOE using the following cipher suites in the Client Hello: - TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection is failing. • Attempt to establish a TLS connection to the TOE using the following cipher suites in the Client Hello: - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection is failing.

	• Attempt to establish a TLS connection to the TOE using the following cipher suites in the Client Hello: - NULL_WITH_NULL_NULL • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection is failing.
Expected Test Results	The TOE rejects TLS connections with the non-supported cipher suites.
Pass/Fail with Explanation	Pass. The TOE rejects TLS connections with the non-supported cipher suites. This meets the testing requirement.

7.4.3 FCS_TLSS_EXT.1.1 TEST #3A

Item	Data
Test Assurance Activity	Modify a byte in the Client Finished handshake message and verify that the server rejects the connection and does not send any application data.
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: Failure to establish a TLS Session shall be audited. Audit log shall reflect the reason for failure.
Test Steps	• Use the 'acumen-tlss tool' to initiate a connection to the TOE and verify the connection fails when a byte is modified in the client finished handshake. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection is failing.
Expected Test Results	When a TLS client connection is initiated with the TOE using 'acumen-tlss tool' such that the Client finished message is modified, the TOE should drop the connection. The packet capture

	shows a decrypt error that confirms that bytes were changed. The logs show a failure connection and terminates the connection.
Pass/Fail with Explanation	Pass. The TOE rejects the connection after receiving the modified Client Handshake message. This meets the test requirements.

7.4.4 FCS_TLSS_EXT.1.1 TEST #3B

Item	Data
Test Assurance Activity	(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'.
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no failures in this test to capture because no failures are expected.
Test Steps	• Initiate a connection to the TOE with acumen-tlss tool as a client. • Verify with packet capture that claimed cipher suite is used to complete handshake. • Verify with packet capture that no Alert with alert level Fatal (2) messages were sent. • Verify with packet capture that 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. • Verify with packet capture that Finished message and confirm that it does not contain unencrypted data by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. • Verify that successful connection is audited and reflected in audit log.
Expected Test Results	The TOE should establish a successful TLS client connection using the 'acumen-tlss' tool and the packet capture should ensure that the finished message is encrypted by specifying that the first three bytes after hexadecimal 16 is not 14.
Pass/Fail with Explanation	Pass. The Finished message contains Hexadecimal 16 and is sent immediately after Hexadecimal 14 in the Change Cipher Spec message. The first byte of the encrypted Finished message does not equal hexadecimal 14. This meets the testing requirement.

7.4.5 FCS_TLSS_EXT.1.2 TEST #1

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: Failure to establish a TLS Session shall be audited. Audit log shall reflect the reason for failure.
Test Steps	• Use the 'acumen-tlss tool' to initiate a connection to the TOE and verify the connections fails except TLSv1.2 • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log.

	• Verify with packet capture that the connection is failing.
Expected Test Results	The TOE should reject the TLS connection that is formed by the 'acumen-tlss tool 'using tls versions below tls v1.2. The packet capture depicts that when the version is less than 1.2, the TOE closes the connection.
Pass/Fail with Explanation	Pass. The TOE rejects all connections except TLS v1.2 connection. This meets the testing requirement.

7.4.6 FCS_TLSS_EXT.1.3 TEST #1A

Item	Data
Test Assurance Activity	If **ECDHE ciphersuites** are supported: 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.
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no failures in this test to capture because no failures are expected.
Test Steps	• Connect to the TOE using secp256r1 and verify that it is successful. • Verify with packet capture that the connection is passing. • Connect to the TOE using secp384r1 and verify that it is successful. • Verify with packet capture that the connection is passing. • Connect to the TOE using secp521r1 and verify that it is successful. • Verify with packet capture that the connection is passing.
Expected Test Results	The TOE should establish a successful TLS connection with all the supported elliptic curves. The packet capture accurately depicts a successful connection with every elliptic curve.
Pass/Fail with Explanation	Pass. The TOE was able to make connection using each supported elliptic curve. This meets the test requirements.

7.4.7 FCS_TLSS_EXT.1.3 TEST #1B

Item	Data
Test Assurance Activity	If **ECDHE ciphersuites** are supported: 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.
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: Failure to establish a TLS Session shall be audited. Audit log shall reflect the reason for failure.
Test Steps	• Connect to the TOE using secp224r1 and verify that it fails. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that the connection is failing.
Expected Test Results	The TOE rejects a TLS client connection when an unsupported elliptic curve is used to establish the session. The packet capture shows that there is an unsuccessful connection and the type of unsupported curve used. The logs confirm that there is a handshake failure.
Pass/Fail with Explanation	Pass. The TOE rejects a connection with unsupported curves. This meets the testing requirements.

7.4.8 FCS_TLSS_EXT.1.3 TEST #2

Item	Data
Test Assurance Activity	If **DHE ciphersuites** are supported, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use a supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the message are the ones configured Diffie-Hellman parameter size(s).
Pass/Fail with Explanation	NA.

	The ST does not claim DHE cipher suites. Hence, this test is not applicable for this TOE.

7.4.9 FCS_TLSS_EXT.1.3 TEST #3

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no failures in this test to capture because no failures are expected.
Test Steps	• Connect to the TOE using RSA 2048 bit key and verify that it is successful. • Verify with packet capture that the connection is passing.
Expected Test Results	The TOE should successfully establish a TLS client connection with both the key sizes and the key size is highlighted in the screenshot. The packet capture shows the key modulus that corresponds to the specific key size thus denoting those successful connections are established with help of both key sizes.
Pass/Fail with Explanation	Pass. The TOE only supports 2048-bit keys for the local certificate. The TOE was able to establish the connection using the supported RSA key size. This meets the testing requirement.

7.4.10 FCS_TLSS_EXT.1.4 TEST #1 [TD0569]

Item	Data
Test Assurance Activity	*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. **TD0569 has been applied.**
Audit Log Requirement	[PP] FCS_TLSS_EXT.1: There are no failures in this test to capture because no failures are expected.
Test Steps	• Use the acumen-tlss tool to initiate a connection to the TOE and verify TOE doesn't set a session ID or ticket. • Verify with packet capture that the connection is passing.
Expected Test Results	TOE (server) makes successful connection with client where client does not send any value other than 0 in session ID and session ticket extension and server hello contains session id value equals to zero.
Pass/Fail with Explanation	Pass.

	TOE does not support session resumption based on session IDs or session ticket. This meets the testing requirements.

7.4.11 FCS_TLSS_EXT.1.4 TEST #2A [TD0569]

Item	Data
Test Assurance Activity	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). 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. **TD0569 has been applied.**
Pass/Fail with Explanation	NA. The TOE does not support session resumption based on session IDs or Session Tickets. Hence, this test is not applicable for this TOE.

7.4.12 FCS_TLSS_EXT.1.4 TEST #2B [TD0569]

Item	Data
Test Assurance Activity	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 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. **TD0569 has been applied.**
Pass/Fail with Explanation	NA. The TOE does not support session resumption based on session IDs or Session Tickets. Hence, this test is not applicable for this TOE.

7.4.13 FCS_TLSS_EXT.1.4 TEST #3A [TD0556, TD0569]

Item	Data
Test Assurance Activity	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. 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. **TD0556 and TD0569 has been applied.**
Pass/Fail with Explanation	NA. The TOE does not support session resumption based on session IDs or Session Tickets. Hence, this test is not applicable for this TOE.

7.4.14 FCS_TLSS_EXT.1.4 TEST #3B [TD0569]

Item	Data
Test Assurance Activity	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 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. **TD0569 has been applied.**
Pass/Fail with Explanation	NA. The TOE does not support session resumption based on session IDs or Session Tickets. Hence, this test is not applicable for this TOE.

7.5 TLSS-MA

7.5.1 FCS_TLSS_EXT.2.1&2 TEST #1A

Item	Data
Test Assurance Activity	If the **TOE requires or can be configured to require a client certificate**, the evaluator shall configure the TOE to require a client certificate and send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify that the handshake is not finished successfully and no application data flows.
Audit Log Requirement	[PP] FCS_TLSS_EXT.2: Failure to authenticate the client shall be audited. Audit log shall reflect the reason for failure.
Test Steps	• Configure the reference identifier on TOE. • Connect using “acumen-tlss” tool by sending the empty certificate_list and show the connection fails. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that connection is failing.
Expected Test Results	The TOE rejects the connection when the client tries to connect with the empty certificate.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when the client tries to connect with the empty certificate_list. This meets the testing requirements.

7.5.2 FCS_TLSS_EXT.2.1&2 TEST #1B

Item	Data
Test Assurance Activity	If the **TOE supports fallback authentication functions and these functions cannot be disabled**. The evaluator shall configure the fallback authentication functions on the TOE and configure the TOE to send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify the TOE authenticates the connection using the fallback authentication functions as described in the TSS.

	Note: Testing the validity of the client certificate is performed as part of X.509 testing.
Pass/Fail with Explanation	NA. This feature does not support as per ST. Hence, this test is not applicable for this TOE.

7.5.3 FCS_TLSS_EXT.2.1&2 TEST #2

Item	**Data**
Test Assurance Activity	If **TLS 1.2 is claimed for the TOE**, the evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify that the connection is denied.
Audit Log requirement	[PP] FCS_TLSS_EXT.2: Failure to authenticate the client shall be audited. Audit log shall reflect the reason for failure.
Test Steps	• Generate a client certificate without supported_signature_algorithm. • The evaluator shall attempt a connection using the client certificate and show the connection being unsuccessful. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that connection is failing.
Expected Test Results	The TOE rejects the client certificate without the supported_signature_algorithm.
Pass/Fail with Explanation	Pass. The TOE rejects mutually authenticated TLS connection attempt from a client containing an unsupported signature algorithm. This meets testing requirements.

7.5.4 FCS_TLSS_EXT.2.1&2 TEST #3

Item	**Data**

Test Assurance Activity	The aim of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To carry out this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognized by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not terminate in the claimed CA certificate). The evaluator shall verify that the attempted connection is denied.
Audit Log requirement	[PP] FCS_TLSS_EXT.2: Failure to authenticate the client shall be audited. Audit log shall reflect the reason for failure.
Test Steps	• Verify the TOE CA details. • Create a CA certificate whose DN matches with the CA certificate on the TOE but with a different key. • Verify that Client certificate is signed by Newly created Impostor certificate (AcumenICA-New-Impostor). • Attempt the connection to the TOE and show the connection fails. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that connection is failing.
Expected Test Results	The TOE denied a connection when it could not verify the validity of the CA in the client certificate.
Pass/Fail with Explanation	Pass. The TOE denied a connection when it could not verify the validity of the CA in the client certificate. This meets testing requirements.

7.5.5 FCS_TLSS_EXT.2.1&2 TEST #4

Item	Data
Test Assurance Activity	The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that the server denies the connection. Ideally, the two certificates should be identical except for the Client Authentication purpose.

Audit Log Requirement	[PP] FCS_TLSS_EXT.2: Failure to authenticate the client shall be audited. Audit log shall reflect the reason for failure.
Test Steps	Valid Certificate: • Load the client certificate containing the Client Authentication purpose. • Initiate a connection with the TOE over TLS and show the connection being successful. • Verify with packet capture that the connection is passing. Invalid Certificate: • Load the client certificate lacking the Client Authentication purpose. • Initiate a connection with the TOE over TLS and show the connection being unsuccessful. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that connection is failing.
Expected Test Results	The TOE makes the successful connection when client presents certificate with client authentication purpose in extended key usage and denies when client certificate lacks the client authentication purpose in extended key usage.
Pass/Fail with Explanation	Pass. The TOE makes the successful connection when client presents certificate with client authentication purpose in extended key usage and denies when client certificate lacks the client authentication purpose in extended key usage. This meets the testing requirements.

7.5.6 FCS_TLSS_EXT.2.1&2 TEST #5A

Item	Data
Test Assurance Activity	Configure the server to require mutual authentication and then connect to the server with a client configured to send a client certificate that is signed by a Certificate Authority trusted by the TOE. The evaluator shall verify that the server accepts the connection.
Audit Log requirement	[PP] FCS_TLSS_EXT.2: There are no failures in this test to capture because no failures are expected.

Test Steps	• Upload a complete certificate validation chain to the TOE. • Initiate a connection with the TOE over TLS and show the connection being successful. • Verify with packet capture that the connection is passing.
Expected Test Results	TOE accepts the connection for the client certificates signed by CA which is trusted by the TOE.
Pass/Fail with Explanation	Pass. TOE accepts the connection for the client certificates signed by CA which is trusted by the TOE. This meets the testing requirements.

7.5.7 FCS_TLSS_EXT.2.1&2 TEST #5B

Item	Data
Test Assurance Activity	Configure the server to require mutual authentication and then modify a byte in the signature block of the client's Certificate Verify handshake message (see RFC5246 Sec 7.4.8). The evaluator shall verify that the server rejects the connection.
Audit Log requirement	[PP] FCS_TLSS_EXT.2: Failure to authenticate the client shall be audited. Audit log shall reflect the reason for failure.
Test Steps	• Use the Acumen TLS modification tool to modify a byte in the client certificate. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that connection is failing.
Expected Test Results	The TOE properly rejects a connection when it receives a modified signature block in the client certificate.
Pass/Fail with Explanation	Pass. The TOE properly rejects a connection when it receives a modified signature block in the client certificate. This meets the testing requirements.

7.5.8 FCS_TLSS_EXT.2.1&2 TEST #6

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSS_EXT.2: There are no failures in this test to capture because no failures are expected.
Test Steps	• Upload a complete certificate validation chain to the TOE. • Initiate a connection with the TOE over TLS and show the connection being successful. • Verify with packet capture that the connection is passing.
Expected Test Results	The TOE allows a certificate to succeed when there is complete certificate validation chain.
Pass/Fail with Explanation	Pass. The TOE also allows a certificate to succeed when there is complete certificate validation chain. This meets testing requirements.

7.5.9 FCS_TLSS_EXT.2.1&2 TEST #7

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FCS_TLSS_EXT.2: Failure to authenticate the client shall be audited. Audit log shall reflect the reason for failure.
Test Steps	This test is performed in conjunction with the FIA_X509_EXT.1.1/Rev Tests.

Expected Test Results	This test is performed in conjunction with the FIA_X509_EXT.1.1/Rev Tests.
Pass/Fail with Explanation	Pass. This test is performed in conjunction with the FIA_X509_EXT.1.1/Rev Tests. The results were found to meet these testing requirements.

7.5.10 FCS_TLSS_EXT.2.1&2 TEST #8

Item	Data
Test Assurance Activity	The purpose of this test is to verify that only selected certificate validation failures could be administratively overridden. **If any override mechanism is defined for failed certificate validation**, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.
Pass/Fail with Explanation	NA. This feature is not claimed in ST. Hence, this test is not applicable for this TOE.

7.5.11 FCS_TLSS_EXT.2.3 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall send a client certificate with an identifier that does not match an expected identifier and verify that the server denies the connection.
Audit Log Requirement	[PP] FCS_TLSS_EXT.2: Failure to authenticate the client shall be audited. Audit log shall reflect the reason for failure.

Test Steps	• Configure the TOE for reference identifier name as FQDN. • Configure the Client certificate (VM) which has mismatched CN. • Initiate the connection to the TLS Server (TOE) with TLS client (VM) and show the connection being unsuccessful. • Verify that the failure of the connection is audited and the reason for failure is reflected in the audit log. • Verify with packet capture that connection is failing.
Expected Test Results	Connection fails when reference identifier does not match the configured identifier.
Pass/Fail with Explanation	Pass. Connection fails when reference identifier does not match the configured identifier. This meets the testing requirements.

7.6 UPDATE

7.6.1 FPT_TST_EXT.1 TEST #1

Item	Data
Test Assurance Activity	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. 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.
Audit Log Requirement	NDcPP FAU_GEN.1.1: Start-up and shut down of the audit functions
Test Steps	• Power on the TOE and observe the TOE Start up. • Ensure that evidence of the execution of self-tests are provided.
Expected Test Results	The TOE executes all required self-tests during bootup.
Pass/Fail with Explanation	Pass. The TOE successfully executes self-test. This meets the testing requirement.

7.6.2 FPT_TUD_EXT.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator performs the version verification activity to determine the current version of the product as well as the most recently installed version (should be the same version before updating).

	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.
Audit Log Requirement	[PP] FPT_TUD_EXT.1: Any attempt to initiate a manual update with the result of the attempt (either success or failure).
Test Steps	• Verify the current operational image details on TOE. • Install the new image provided by the vendor. • Set the new image to be the operational image of the TOE. • Verify that the current operational image is changed to the new image. • Verify with the audit logs that this attempt was recorded.
Expected Test Results	The TOE successfully updates the current image version with the new image after verifying that the new image is authentic. The logs indicate the same that the new image is verified and has then been installed.
Pass/Fail with Explanation	Pass. The TOE successfully upgraded with new build. This meets the testing requirement.

7.6.3 FPT_TUD_EXT.1 TEST #2 (A)

Item	Data
Test Assurance Activity	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 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.
Audit Log Requirement	[PP] FPT_TUD_EXT.1: Any attempt to initiate a manual update with the result of the attempt (either success or failure).
Test Steps	• Verify the current operational image details of the TOE. • Upload an image of a legitimately signed image that has a modified bit (an image with a valid signature but have a few bits off than a legitimate image). • Attempt to install the modified image and verify that it is failing. • Verify that an audit log for the attempt to install the image is emitted and the result of the attempt is present in the audit log. • Verify that the current operational image details on the TOE remains the same. • Verify that the corrupted image cannot be selected as the next boot image. •
Expected Test Results	TOE was unable to successfully upgrade current image with modified image file as TOE has capabilities to identified changed bit in modified image file which retain integrity of image file.
Pass/Fail with Explanation	Pass. The TOE software was able to detect when an image was corrupted and rejected the image. This meets the testing requirements.

7.6.4 FPT_TUD_EXT.1 TEST #2 (B)

Item	Data
Test Assurance Activity	[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: 2) An image that has not been signed 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.
Audit Log Requirement	[PP] FPT_TUD_EXT.1: Any attempt to initiate a manual update with the result of the attempt (either success or failure).
Test Steps	• Verify the current operational image details of the TOE. • Upload an image without a signature (an image with signature bits stripped off). • Attempt to install the modified image and verify that it is failing. • Verify that an audit log for the attempt to install the image is emitted and the result of the attempt is present in the audit log. • Verify that the current operational image details on the TOE remains the same. • Verify that the corrupted image cannot be selected as the next boot image.
Expected Test Results	TOE can successfully check and verify the signature of image file. Hence, TOE has not allowed to upgrade image without required signature in image file.
Pass/Fail with Explanation	Pass. The TOE software was able to detect when an image was not signed and rejected the image. This meets the testing requirements.

7.6.5 FPT_TUD_EXT.1 TEST #2 (C)

Item	Data

Test Assurance Activity	[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: 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) 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.
Audit Log Requirement	[PP] FPT_TUD_EXT.1: Any attempt to initiate a manual update with the result of the attempt (either success or failure).
Test Steps	• Verify the current operational image details of the TOE. • Upload an image that has an invalid signature (an image which have a few signature bits not matching a legitimate signature). • Attempt to install the modified image and verify that it is failing. • Verify that an audit log for the attempt to install the image is emitted and the result of the attempt is present in the audit log. • Verify that the current operational image details on the TOE remains the same. • Verify that the corrupted image cannot be selected as the next boot image.
Expected Test Results	TOE can successfully check and verify the signature of image file. Hence, TOE has not allowed to upgrade image without required signature in image file.
Pass/Fail with Explanation	Pass. The TOE software was able to detect when an image had an invalid signature and rejected the image. This meets the testing requirements.

7.6.6 FPT_TUD_EXT.1 TEST #3 (A)

Item	Data
Test Assurance Activity	[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. 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 Security Administrator 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 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.
Pass/Fail with Explanation	NA. TOE does not verify has value over an image against published has value. Instead, it uses digital signature as per ST. Hence, this test is not applicable for this TOE.

7.6.7 FPT_TUD_EXT.1 TEST #3 (B)

Item	Data
Test Assurance Activity	[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. 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 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.

Pass/Fail with Explanation	NA. TOE does not verify has value over an image against published has value. Instead, it uses digital signature as per ST. Hence, this test is not applicable for this TOE.

7.6.8 FPT_TUD_EXT.2 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall verify that the update mechanism includes a certificate validation according to FIA_X509_EXT.1 and a check for the Code Signing purpose in the extendedKeyUsage. The evaluator shall digitally sign the update with an invalid certificate and verify that update installation fails. The evaluator shall digitally sign the application with a certificate that does not have the Code Signing purpose and verify that application installation fails. The evaluator shall repeat the test using a valid certificate and a certificate that contains the Code Signing purpose and verify that the application installation succeeds. The evaluator shall use a previously valid but expired certificate and verifies that the TOE reacts as described in the TSS and the guidance documentation. Testing for this element is performed in conjunction with the assurance activities for FPT_TUD_EXT.1 The evaluator shall demonstrate that checking the validity of a certificate is performed at the time a certificate is used when performing trusted updates. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device.
Pass/Fail with Explanation	NA. This function is not claimed in ST. Hence, this test is not applicable for this TOE.

7.7 X509

7.7.1 FIA_X509_EXT.1.1/REV TEST #1A

Item	Data
Test Assurance Activity	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).
Audit Log Requirement	[PP] FIA_X509_EXT.1: Unsuccessful attempt to validate a certificate shall be audited with the reason for failure of certificate validation. [PP] FIA_X509_EXT.1: When certificates are added/deleted/replaced, an audit log shall be generated. Audit log shall identify the certificate that got updated.
Test Steps	When the full certificate chain is present on the TOE: • Create a full chain of certificates to connect to the TOE. • Upload a complete certificate validation chain to the TOE. • Attempt to connect to the TOE with only the server certificate sent in the wire. • Verify with packet capture that the connection is successfully established, and that application data is flowing. When the Intermediate CA is missing in the TOE but is sent to the TOE by the server: • Remove the existing CA-Chain from the TOE and upload a CA-Chain where only the RootCA matches the chain presented by the peer. • Attempt to connect to the TOE by sending the server certificate and the Intermediate CA along the wire by the TLS Server. • Verify with packet capture that the connection is successfully established, and that application data is flowing.
Expected Test Results	The TOE establishes a TLS server connection successfully when it is provided with a complete chain of certificates. The packet capture shows that a successful connection has been established and it provides the entire chain of certificates.
Pass/Fail with Explanation	Pass.

	When a complete certificate trust chain is present during a connection, the TOE can make a successful connection. This meets the testing requirements.

7.7.2 FIA_X509_EXT.1.1/REV TEST #1B

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	[PP] FIA_X509_EXT.1: Unsuccessful attempt to validate a certificate shall be audited with the reason for failure of certificate validation. [PP] FIA_X509_EXT.1: When certificates are added/deleted/replaced, an audit log shall be generated. Audit log shall identify the certificate that got updated.
Test Steps	• Modify the TOE Certificate Trust Store to have a broken chain by removing the Intermediate CA. • Verify that the audit logs are present for the management of the certificates. • Attempt to connect to the TOE with a server certificate with an incomplete chain and verify that it fails. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	When a complete certificate chain is not provided, the TOE fails to establish a TLS server connection. The packet capture shows that this connection is not established due to an unknown CA certificate. The logs provide concrete evidence that states the fact that the TOE is unable to retrieve the local issuer certificate.
Pass/Fail with Explanation	Pass. When an incomplete certificate trust chain is present, the TOE is not able to make a successful connection. This meets the testing requirements.

7.7.3 FIA_X509_EXT.1.1/REV TEST #2

Item	Data
Test Assurance Activity	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. Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.
Audit Log Requirement	[PP] FIA_X509_EXT.1: Unsuccessful attempt to validate a certificate shall be audited with the reason for failure of certificate validation. [PP] FIA_X509_EXT.1: When certificates are added/deleted/replaced, an audit log shall be generated. Audit log shall identify the certificate that got updated.
Test Steps	• Create a server certificate which is expired. • Attempt to connect to the TOE with the expired server certificate and verify that it fails. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful. • Create an Intermediate CA certificate about to expire in 4Hrs. and upload it on TOE. • Attempt to connect to the TOE with non-expired server certificate and verify that connection is successful. • Verify with packet capture that the connection is successfully established, and that application data is flowing. • Change the time on TOE to make Intermediate CA certificate act as an Expired Intermediate CA certificate. • Attempt to connect to the TOE with non-expired server certificate and verify that connection is not successful. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The TOE rejects the TLS server connection because the certificate has expired (for both leaf certificate and intermediate CA). The packet capture confirms that the connection wasn't established and shows when the certificate has expired. The logs attest to the fact that the certificate has expired.
Pass/Fail with Explanation	Pass.

	The TOE denied the connection when an expired certificate (Leaf Certificate and Intermediate Certificate) is found within the connection request. This meets the testing requirements.

7.7.4 FIA_X509_EXT.1.1/REV TEST #3

Item	Data
Test Assurance Activity	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. 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.
Audit Log Requirement	[PP] FIA_X509_EXT.1: Unsuccessful attempt to validate a certificate shall be audited with the reason for failure of certificate validation. [PP] FIA_X509_EXT.1: When certificates are added/deleted/replaced, an audit log shall be generated. Audit log shall identify the certificate that got updated.
Test Steps	• C Create server certificate. • Create Intermediate CA certificate with CRL Signing enabled. • Import the CA certificates chain on the TOE. • Attempt a connection and verify that it is successful. • Verify with logs. • Verify with packet capture that the connection is successfully established, and that application data is flowing. • Revoke the server certificate. • Attempt a connection with the TOE and verify that it fails.

	• Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful. • Revoke the intermediate CA certificate. • Verify that the database shows that certificate is revoked. • Attempt a connection with the TOE and verify that it fails. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The TOE rejects any TLS server connection when either the intermediate certificate or the server certificate has been revoked. The CRL connection also shows that the certificates have been revoked. The Packet capture depicts the specific certificate that has been revoked and the logs verify that the TOE has denied connection by denoting that certificate has been revoked.
Pass/Fail with Explanation	Pass. Connection with revoked certificate is not accepted by the TOE which meet the requirement.

7.7.5 FIA_X509_EXT.1.1/REV TEST #4

Item	Data
Test Assurance Activity	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. 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.
Audit Log Requirement	[PP] FIA_X509_EXT.1: Unsuccessful attempt to validate a certificate shall be audited with the reason for failure of certificate validation. [PP] FIA_X509_EXT.1: When certificates are added/deleted/replaced, an audit log shall be generated. Audit log shall identify the certificate that got updated.

Test Steps	• Generate an Intermediate CA certificate that does NOT have CRL signing Key Usage. • Upload generated Intermediate CA certificate to the TOE Trust Store, and it fails. • Verify that uploading generated Intermediate CA certificate to the TOE Trust Store fail, and the failure is audited with the reason for failure reflected in the audit log. • Upload valid Root CA and Intermediate CA (Imposter_ICA) certificate that is not used to sign the server leaf certificate. • Attempt to make a connection to the TOE with an Intermediate CA certificate that doesn't have CRL SIGN key usage and verify it fails. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The TOE doesn't allow to upload CA chain certificate when the CRL signing purpose is missing and validation fails. The packet capture shows that there is a handshake failure due to the absence of CRL Signing. The logs are used to validate the fact that the connection has been rejected by CRL due to failure in certificate verification.
Pass/Fail with Explanation	Pass. The TOE rejects connections when the Signer certificate does not have CRL signing parameter in Key Usage. This meets requirements.

7.7.6 FIA_X509_EXT.1.1/REV TEST #5

Item	Data
Test Assurance Activity	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. 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.)
Audit Log Requirement	[PP] FIA_X509_EXT.1: • Unsuccessful attempt to validate a certificate shall be audited with the reason for failure of certificate validation. • When certificates are added/deleted/replaced, an audit log shall be generated. Audit log shall identify the certificate that got updated.
Test Steps	• Attempt a connection to a remote modified TLS server using 'acumen-tlsc-v2.2e tool' that would perform the necessary modification (modify first 8 bytes of server certificate) on the server certificate. Verify that the TOE rejects the connection.

	• Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The TOE denies a TLS connection when it is presented with a certificate that has been modified using the 'acumen-tlsc-v2.2e tool'. The tool modifies the first eight bytes of the certificate. The packet capture verifies that the connection is not established due to the bad certificate. The logs depict that there's an encoding error thus verifying that the connection was rejected.
Pass/Fail with Explanation	Pass. The TOE rejects connections when the first 8 bytes of the certificate are modified. This meets the testing requirements.

7.7.7 FIA_X509_EXT.1.1/REV TEST #6

Item	Data
Test Assurance Activity	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. 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.)
Audit Log Requirement	[PP] FIA_X509_EXT.1: Unsuccessful attempt to validate a certificate shall be audited with the reason for failure of certificate validation. [PP] FIA_X509_EXT.1: When certificates are added/deleted/replaced, an audit log shall be generated. Audit log shall identify the certificate that got updated.
Test Steps	• Attempt a connection to a remote TLS server with a modified certificate (last byte in certificate) using 'acumen-tlsc-v2.2e tool' and verify that it fails. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The TOE fails to establish a TLS connection when the last byte in the signature Value field of the certificate is modified using the 'acumen-tlsc-v2.2e tool'. The packet capture proves that there is a decrypt error, and the logs show that there is a failure in establishing connection due to certificate signature failure.

Pass/Fail with Explanation	Pass. The modified certificate (last byte in certificate) fails to validate. This meets the testing requirement.

7.7.8 FIA_X509_EXT.1.1/REV TEST #7

Item	Data
Test Assurance Activity	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. 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.)
Audit Log Requirement	[PP] FIA_X509_EXT.1: Unsuccessful attempt to validate a certificate shall be audited with the reason for failure of certificate validation. [PP] FIA_X509_EXT.1: When certificates are added/deleted/replaced, an audit log shall be generated. Audit log shall identify the certificate that got updated.
Test Steps	• Attempt a connection to a remote TLS server using 'acumen-tlsc-v2.2e tool' and modify any byte in the public key of the certificate and verify that the connection is rejected. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The TOE rejects a remote TLS connection that is formed using the 'acumen-tlsc-v2.2e tool'. The tool modifies the certificate such that its public key is modified and uses the same certificate for establishing the TLS connection. The packet capture depicts that there is a decrypt error, and the logs show a failure in establishing a connection due to certificate signature failure.
Pass/Fail with Explanation	Pass. The TOE rejects a connection when the bytes inside the public key of the certificate are modified. This meets the testing requirement.

7.7.9 FIA_X509_EXT.1.1/REV TEST #8A [TD0527]

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** **(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. **TD0527 (12/1 Update) has been applied.**
Pass/Fail with Explanation	NA. This feature is not claimed in ST. Hence, this test is not applicable for this TOE.

7.7.10 FIA_X509_EXT.1.1/REV TEST #8B

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** **(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. **TD0527 (12/1 Update) has been applied.**
Pass/Fail with Explanation	NA. This feature is not claimed in ST. Hence, this test is not applicable for this TOE.

7.7.11 FIA_X509_EXT.1.1/REV TEST #8C

Item	Data
Test Assurance Activity	**(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen)** **(Conditional on support for a minimum certificate path length of three certificates)** 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. **TD0527 (12/1 Update) has been applied.**
Pass/Fail with Explanation	NA. This feature is not claimed in ST. Hence, this test is not applicable for this TOE.

7.7.12 FIA_X509_EXT.1.2/REV TEST #1

Item	Data
Test Assurance Activity	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). 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) (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).*
Audit Log Requirement	[PP] FIA_X509_EXT.1: Unsuccessful attempt to validate a certificate shall be audited with the reason for failure of certificate validation. [PP] FIA_X509_EXT.1: When certificates are added/deleted/replaced, an audit log shall be generated. Audit log shall identify the certificate that got updated.
Test Steps	• Create a new Intermediate CA certificate without basicConstraints. • Concatenate the Root CA certificate and modified Intermediate CA certificate. • Attempt to upload concatenate file on TOE and verify it is failed. • Verify that uploading generated Intermediate CA certificate to the TOE Trust Store fail, and the failure is audited with the reason for failure reflected in the audit log. • Upload Root CA + Imposter Intermediate CA on the TOE and verify it is successfully uploaded on TOE. • Verify that uploading generated Root CA + Imposter Intermediate CA to the TOE Trust Store is successful and an audit log is generated for the event. • Attempt the connection from the TOE to the TLS Server by sending Intermediate CA certificate (without basicConstraints). • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.

Expected Test Results	The TOE rejects the connection request with an intermediate CA certificate which does not contain the basicConstraints extension. The packet capture depicts that an unknown CA had been used. The logs show a failure in establishing connection as the verification of certificate failed. The TOE also rejects to accept CA certificates without basicConstraints when uploading being uploaded into the TOE's certificate trust store.
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that doesn't contain the basicConstraints extension. This meets the test requirements.

7.7.13 FIA_X509_EXT.1.2/REV TEST #2

Item	Data
Test Assurance Activity	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 it to verify that the TOE accepts only certificates that have been marked as CA certificates by using basicConstraints with the CA flag set to True (and implicitly 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). 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: 1. As part of the validation of the leaf certificate belonging to this chain; 2. 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).

Audit Log Requirement	[PP] FIA_X509_EXT.1: Unsuccessful attempt to validate a certificate shall be audited with the reason for failure of certificate validation. [PP] FIA_X509_EXT.1: When certificates are added/deleted/replaced, an audit log shall be generated. Audit log shall identify the certificate that got updated.
Test Steps	• Make CA flag FALSE in Intermediate CA certificate via "x509-mod" tool. • Show modified flag in Intermediate CA certificate. • Attempt to upload concatenate file on TOE and verify it is failed. • Verify that uploading generated concatenate file to the TOE Trust Store fail, and the failure is audited with the reason for failure reflected in the audit log. • Upload Root CA + Imposter Intermediate CA on the TOE and verify it is successfully uploaded on TOE. • Verify that uploading generated Root CA + Imposter Intermediate CA to the TOE Trust Store is successful and an audit log is generated for this event. • Make connection between TOE and TLS Server. • Verify that the connection attempt fails, and the failure of the connection is audited with the reason for failure reflected in the audit log. • Verify with packet capture that the connection attempt is unsuccessful.
Expected Test Results	The TOE rejects the TLS server connection which uses a CA certificate that has been modified using the 'x509-mod tool' such that the CA certificate contains basicConstraints 'CA is set to false'. The packet capture shows that the 'basicConstraints for CA' is set to false, and the logs show a failure in establishing a connection due to use of an invalid CA certificate. The TOE also rejects to accept CA certificates with basicConstraints 'CA is set to false' when uploading being uploaded into the TOE's certificate trust store.
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by an ICA that has the CA flag in the basicConstraints extension set to FALSE. This meets the test requirements.

7.7.14 FIA_X509_EXT.2 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each trusted channel: The evaluator shall demonstrate that using a valid certificate 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.
Audit Log Requirement	[PP] FIA_X509_EXT.2, there are no audit logs required. However, Application Note 59 in the PP states that failures to read or access the CRL (or OCSP responder) should be audited. Hence, there is an implicit audit message requirement for failing to validate the certificate using the CRL or OCSP component.
Test Steps	• Create a server certificate with a CRL distribution point and modified CRL filename. • Start the CRL Server on the VM (10.1.5.44) as the remote TLS Server (10.1.5.44) which does not match the CRL server IP on the certificate. • Attempt to connect to the TOE using Openssl and verify that it passes. • Verify that the connection attempt passes, and the success of the connection is audited with the reason for success reflected in the audit log. • Verify with packet capture that the connection is successfully established, and that application data is flowing.
Expected Test Results	When the TOE cannot establish a connection to determine the validity of a certificate, the TOE should still accept the certificate. The packet capture should show that the connection is successfully established.
Pass/Fail with Explanation	Pass. The TOE still accepts certificates although it cannot reach the CRL responder. This meets the testing requirements.

7.7.15 FIA_X509_EXT.3 TEST #1

Item	Data

Test Assurance Activity	The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.
Audit Log Requirement	Because this test invokes creating a private key, FAU_GEN.1.1(c), bullet point 3 requires logging of the action and the associated identifier.

Test Steps	• From the TOE, generate a CSR.
	• Examine the CSR contents and ensure the CSR contains the following fields.
	  ○ Public Key
	  ○ Common Name.
	  ○ Organization.
	  ○ Organizational Unit.
	  ○ Country.

Expected Test Results	The TOE will successfully generate a CSR with the help of an RSA key.
Pass/Fail with Explanation	Pass. The TOE can generate CSR with all the requisite information. This meets the testing requirements.

7.7.16 FIA_X509_EXT.3 TEST #2

Item	Data
Test Assurance Activity	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.
Audit Log Requirement	Because this test invokes creating a private key, FAU_GEN.1.1(c), bullet point 3 requires logging of the action and the associated identifier.
Test Steps	• From the TOE, generate a CSR request. • Generate a signed certificate based on the generated CSR from an external CA. • Ensure that the full trust chain for the signed CA is not present on the TOE. • Attempt to load the signed certificate on the TOE.

	• Verify to upload TOE leaf certificate to the TOE fails, and the failure is audited with the reason for failure reflected in the audit log. • Add the intermediary certificates to the TOE certificate store to ensure that the signing CA now has a full certificate path. • Re-attempt to load the signed certificate on the TOE.
Expected Test Results	The TOE doesn't validate a signed CSR if the full trust chain is not present. When a full trust chain is present, the TOE validates the signed CSR.
Pass/Fail with Explanation	Pass. The TOE allows a certificate to be installed when the complete trust chain is present and rejects a certificate when the complete trust chain is not present. This meets the testing requirement.

8 CAVP MAPPING

8.1 TOE MODELS AND CRYPTOGRAPHIC OPERATIONAL ENVIRONMENT

This section presents a detailed listing of each card supplying cryptographic functionality and its associate cryptographic operational environment (OE).

TOE Card/Model	Operating Environment
NATX-8-100G-CC NATX-16-100G-CC NATX-32-100G-1-CC NATX-64-100G-2-CC MMA10G-NATX-8-CC MMA10G-NATX-16-CC MMA10G-NATX-32-CC MMA10G-NATX-64-CC MMA10G-IPX128 3080IPX-48-25G-CC	Customized Linux 4.19 on Intel(R) Core (TM) i3-6102E CPU

8.2 OPERATIONAL ENVIRONMENT OF THE ALGORITHM IMPLEMENTATION

This section presents a detailed listing of each algorithm listing to include the name and the OE.

Algorithm	Cert #	Name	Operating Environment
AES	A2573	EXE Cryptographic Module	Customized Linux 4.19 on Intel(R) Core (TM) i3-6102E CPU
SHS	A2573	EXE Cryptographic Module	Customized Linux 4.19 on Intel(R) Core (TM) i3-6102E CPU
HMAC	A2573	EXE Cryptographic Module	Customized Linux 4.19 on Intel(R) Core (TM) i3-6102E CPU
DRBG	A2573	EXE Cryptographic Module	Customized Linux 4.19 on Intel(R) Core (TM) i3-6102E CPU
RSA	A2573	EXE Cryptographic Module	Customized Linux 4.19 on Intel(R) Core (TM) i3-6102E CPU
ECDSA	A2573	EXE Cryptographic Module	Customized Linux 4.19 on Intel(R) Core (TM) i3-6102E CPU
KAS	A2573	EXE Cryptographic Module	Customized Linux 4.19 on Intel(R) Core (TM) i3-6102E CPU

8.3 CERTIFICATE(S) TABLE

This section provides a table that lists all SFRs for which a CAVP certificate is claimed, the CAVP algorithm list name and the CAVP Certificate number.

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	EXE Cryptographic Module	RSA KeyGen (FIPS186-4)	A2573
	ECC schemes using "NIST curves" [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	EXE Cryptographic Module	ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4)	A2573
FCS_CKM.2	RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"	EXE Cryptographic Module	No NIST CAVP, CCTL has performed all assurance/evaluation activities and documented in the ETR and AAR accordingly.	N/A. This testing was performed in conjunction with FTP_TRP.1/Admin Test #1 and FTP_ITC.1 Test #1 to demonstrate correct operation.
	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	EXE Cryptographic Module	KAS-ECC-SSC Sp800-56Ar3	A2573
FCS_COP.1/ DataEncryption	AES used in [**CBC, CTR, GCM**] mode and cryptographic key sizes [128 bits, 256 bits]	EXE Cryptographic Module	AES-CBC AES-CTR AES-GCM	A2573
FCS_COP.1/ SigGen	RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [**2048 bits, 3072 bits, 4096 bits**] For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	EXE Cryptographic Module	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	A2573
FCS_COP.1/ Hash	[**SHA-1, SHA-256, SHA-384**] and message digest sizes [**160, 256, 384**] bits	EXE Cryptographic Module	SHA-1 SHA2-256 SHA2-384	A2573
FCS_COP.1/ KeyedHash	[**HMAC-SHA-1, HMAC-SHA- 256, HMAC-SHA-384**] and cryptographic key sizes [**key size (in bits) used in HMAC**] and message digest sizes [**160, 256, 384**] bits	EXE Cryptographic Module	HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384	A2573
FCS_RBG_EXT.1	**CTR_DRBG (AES)**	EXE Cryptographic Module	Counter DRBG	A2573

9 CONCLUSION

The testing shows that all test cases required for conformance have passed testing.