# MMA10G-EXE Series

# Security Target

Acumen Security, LLC.

Document Version: 1.1

**Prepared For:**
Evertz Microsystems, Ltd.
5292 John Lucas Drive
Burlington, Ontario, CANADA

Prepared by:
Acumen Security
2400 Research Blvd
Rockville MD 20850

## *Table Of Contents*

# Revision History

| Version | Date | Description |
|---|---|---|
| 0.1 | November 12, 2019 | Initial Draft |
| 0.2 | January 21, 2020 | Addressed validator comments |
| 0.3 | January 27, 2020 | Addressed validator comments |
| 1.0 | March 6, 2020 | Final review |
| 1.1 | May 5, 2020 | Addressed validator comments |

# 1 Security Target Introduction

## 1.1 Security Target and TOE Reference

This section provides information needed to identify and control this ST and its TOE.

| Category | Identifier |
|---|---|
| ST Title | MMA10G-EXE Security Target |
| ST Version | 1.1 |
| ST Date | May 5, 2020 |
| ST Author | Acumen Security, LLC. |
| TOE Identifier | MMA10G-EXE Series |
| TOE Hardware | MMA10G-EXE16, MMA10G-EXE26, MMA10G-EXE36, MMA10G-IPX128, EXE2.0-16-10G-A1, EXE2.0-16-25G-A1, EXE2.0-26-10G-A1, EXE2.0-26-25G-A1, EXE2.0-36-10G-A1, EXE2.0-36-25G-A1 |
| TOE Software Version | V1.2 |
| TOE Developer | Evertz Microsystems Ltd.<br>5292 John Lucas Drive<br>Burlington, Ontario<br>CANADA |
| Key Words | Network Device |

**Table 1 TOE/ST Identification**

## 1.2 TOE Overview

### 1.2.1 TOE Product Type

The TOE is classified as a network device (a generic infrastructure device that can be connected to a network). The TOE hardware devices are the Evertz MMA10G-EXE16 (16RU), MMA10G-EXE26 (26RU), MMA10G-EXE36 (36RU), MMA10G-IPX128 (6RU), EXE2.0-16-10G-A1 (16RU), EXE2.0-16-25G-A1 (16RU), EXE2.0-26-10G-A1 (26RU), EXE2.0-26-25G-A1 (26RU), EXE2.0-36-10G-A1 (36RU), EXE2.0-36-25G-A1 (36RU) running EXE v1.2 and will be referred to as EXE throughout this document. The EXE appliances are Ethernet switches optimized for video content.

### 1.2.2 TOE Usage

The MMA10G-EXE switches are 10 Gigabit (Gb) Internet Protocol (IP) switches optimized for video-over-IP traffic (compressed or uncompressed), while the EXE2.0 switches are 25Gb IP switches optimized for video-over-IP traffic. The ten 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. These differences are detailed in Table 3.

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. The TOE exclusively uses multicast IP addressing. Unicast is not supported by the EXE platform.

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 deployment configuration of the TOE is shown in the following figure:

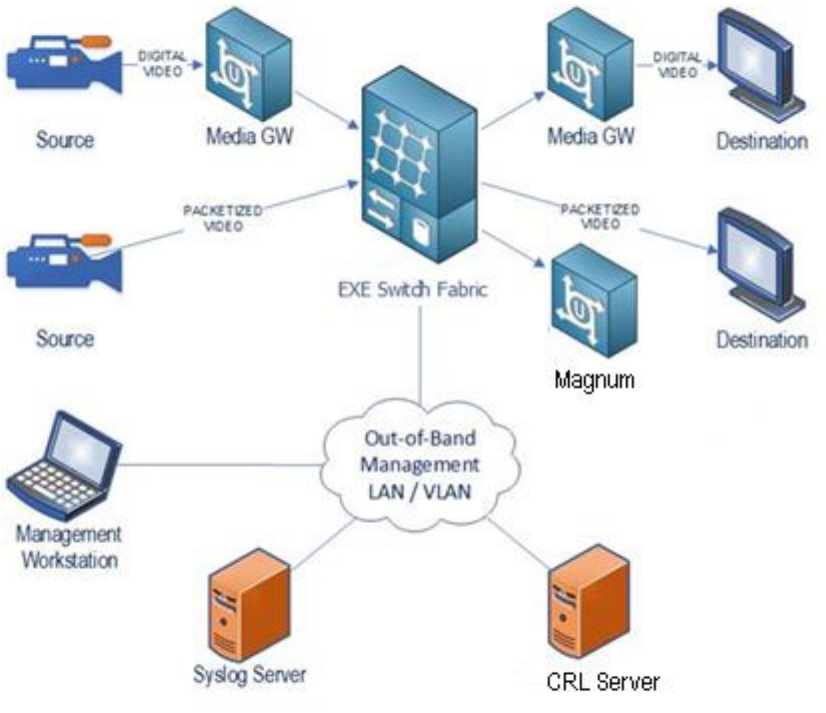

**Figure 1 TOE Topology**[1]

## 1.3 TOE IT Environment

The EXE deploys a Layer 2 switching fabric optimized for streaming compressed and uncompressed video over IP. As a Layer 2 technology, the EXE architecture requires a dedicated physically or logically isolated (“closed”) network which must support IPv4 multicast addressing. Sources and destinations (the nature and format of which are independent of the EXE switching system) interface with the switch fabric via Ethernet, either via direct packetized streams or via media gateways. The EXE performs no signal processing and is format-neutral, allowing support for any IP-compliant broadcast video streaming product.

[1] The TOE boundary is that of the physical Evertz EXE Switch Fabric device as indicated in the figure. All other components are part of the TOE environment.

The TOE's operational environment must provide the following services to support the secure operation of the TOE:

| Component | Required | Usage/Purpose Description for TOE performance |
|---|---|---|
| Syslog server | Yes | • Conformant with RFC 5424 (Syslog Protocol)<br>• Supporting Syslog over TLS (RFC 5425)<br>• Acting as a TLSv1.2 server<br>• Supporting Client Certificate authentication<br>• Supporting at least one of the following cipher suites:<br>○ TLS_RSA_WITH_AES_128_CBC_SHA<br>○ TLS_RSA_WITH_AES_256_CBC_SHA<br>○ TLS_RSA_WITH_AES_128_CBC_SHA256<br>○ TLS_RSA_WITH_AES_256_CBC_SHA256<br>○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256<br>○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 |
| Management laptop | Yes | • Internet Explorer, Google Chrome, or Firefox<br>• Supporting TLSv1.2<br>• Supporting at least one of the following ciphersuites:<br>○ TLS_RSA_WITH_AES_128_CBC_SHA<br>○ TLS_RSA_WITH_AES_256_CBC_SHA<br>○ TLS_RSA_WITH_AES_128_CBC_SHA256 as defined<br>○ TLS_RSA_WITH_AES_256_CBC_SHA256<br>○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256<br>○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 |
| CRL Server | Yes | • Conformant with RFC 5280 |
| Magnum | Yes | • Video switch controller used for management of the TOE |
| Media Gateway | No | • Optional component for converting media streams. Not required for TOE operations |
| Video Source devices | No | • Optional component for creating video streams that are sent to the TOE. Not required for TOE operations.<br>• Supporting packetized or digital video streams. |
| Video Destination devices | No | • Optional component for creating video streams that are sent to the TOE. Not required for TOE operations.<br>• Supporting packetized or digital video streams. |

**Table 2 IT Environment Components**

## 1.4 TOE Architecture

### 1.4.1 Physical Boundaries

Evertz EXE is a 10GbE or 25GbE switch designed for video transport over IP. Chassis can support up to 2304 10GbE ports, with a total switching capacity of 46 Tb/s.

EXE's internal frame controllers provide connectivity to remote control panels and 3rd party control devices such as automation systems via Ethernet ports.

| FRAME | RACK UNITS | INTERFACES | MAXIMUM PORTS | CPU |
|---|---|---|---|---|
| MMA10G-EXE16 | 16 | 16 x QSFP28 cages per line card | 512 | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| MMA10G-EXE26 | 26 | 16 x QSFP28 cages per line card | 1024 | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |

| | | | | |
|---|---|---|---|---|
| MMA10G-EXE36 | 36 | 16 x QSFP28 cages per line card | 2048 | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| MMA10G-IPX128 | 6 | 128 10GE/1GE QSFP+ | 128 | Intel$^{(R)}$ Core $^{(TM)}$ i3-4102E C (Haswell) |
| EXE2.0-16-10G-A1 | 16 | 16 x QSFP28 cages per line card | 512 | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| EXE2.0-16-25G-A1 | 16 | 16 x QSFP28 cages per line card | 512 | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| EXE2.0-26-10G-A1 | 26 | 16 x QSFP28 cages per line card | 1024 | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| EXE2.0-26-25G-A1 | 26 | 16 x QSFP28 cages per line card | 1024 | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| EXE2.0-36-10G-A1 | 36 | 16 x QSFP28 cages per line card | 2048 | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| EXE2.0-36-25G-A1 | 36 | 16 x QSFP28 cages per line card | 2048 | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |

**Table 3 EXE Models**

### 1.4.2 Logical Scope of the TOE

The TOE supports the following functionality:

- Security Audit
- Cryptographic Support
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

### 1.4.3 Security Functions provided by the TOE

The TOE provides the security functionality required by NDcPP v2.1.

#### 1.4.3.1 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 previous audit records are overwritten when the allocated space for these records reaches the threshold on a FIFO basis.

#### 1.4.3.2 Cryptographic Support

The TOE provides cryptography support for secure communications and protection of information. The cryptographic services provided by the TOE include: symmetric encryption and decryption using AES; asymmetric key generation; cryptographic key establishment using ECDH key establishment; digital signature using RSA; cryptographic hashing using SHA-256; random bit generation using DRBG and keyed-hash message authentication using HMAC-SHA (SHA-256). The TOE implements the secure protocols TLS/HTTPS on the server side and TLS on the client side. The algorithm certificate references are listed in the table below.

| Cryptographic Protocol | Use within the TOE |
|---|---|
| HTTPS/TLS (client) | Secure connection to syslog<br>FCS_HTTPS_EXT.1, FCS_TLSC_EXT.1 |
| HTTPS/TLS (server) | Remote management and secure connection to MAGNUM control system<br>FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1 |
| AES | Provides encryption/decryption in support of the TLS protocol.<br>FCS_TLSC_EXT.1, FCS_TLSS_EXT.1 |
| DRBG | Deterministic random bit generation use to generate keys.<br>FCS_TLSS_EXT.1, FCS_RBG_EXT.1 |
| Secure hash | Used as part of digital signatures, firmware integrity checks.<br>FCS_COP.1/Hash, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1 |
| HMAC | Provides keyed hashing services in support of TLS.<br>FCS_COP.1/KeyedHash, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1 |
| EC-DH | Provides key establishment for TLS.<br>FCS_CKM.2, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1 |
| RSA | Provide key generation and signature generation and verification (PKCS1_V1.5) in support of TLS.<br>FCS_CKM.1, FCS_COP.1/SigGen, FCS_COP.1/SigVer, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1 |

**Table 4 TOE Cryptographic Protocols**

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.

| Algorithm | Standard | CAVP Certificate # | Processors |
|---|---|---|---|
| AES 128/256-bit CBC, GCM | IOS 19772 (GCM) | C1448 | Intel$^{(R)}$ Core $^{(TM)}$ i3-4102E C (Haswell) |
| | | | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| CTR DRBG using AES 256 | ISO/IEC 18031:2011 | C1448 | Intel$^{(R)}$ Core $^{(TM)}$ i3-4102E C (Haswell) |
| | | | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| EC-DH | NIST SP 800-56A (key establishment) | C1448 | Intel$^{(R)}$ Core $^{(TM)}$ i3-4102E C (Haswell) |
| | | | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| ECDSA | FIPS PUB 186-4 (key generation) | C1448 | Intel$^{(R)}$ Core $^{(TM)}$ i3-4102E C (Haswell) |
| | | | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| HMAC-SHA-1/256/384 | ISO/IEC 9797-2:2011 | C1448 | Intel$^{(R)}$ Core $^{(TM)}$ i3-4102E C (Haswell) |
| | | | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| SHA-1/256/384 | ISO/IEC 10118-3:2004 | C1448 | Intel$^{(R)}$ Core $^{(TM)}$ i3-4102E C (Haswell) |
| | | | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |
| RSA 2048/3072 | FIPS PUB 186-4 (key generation and Digital Signature)<br>ISO/IEC 9796-2 (digital signature) | C1448 | Intel$^{(R)}$ Core $^{(TM)}$ i3-4102E C (Haswell) |
| | | | Intel$^{(R)}$ Xeon$^{(R)}$ E3-1505M v5 (Skylake) |

**Table 5 CAVP Algorithm Testing References**

#### 1.4.3.3 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's 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 an attempt 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.

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

#### 1.4.3.4 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 cryptographic services
- Update the TOE and verify the updates using digital signature capability prior to installing those updates
- Specify the time limits of session inactivity

All of these management functions are restricted to an Administrator, which covers all administrator roles. Administrators are individuals who manage specific type of administrative tasks. In EXE only the admin role exists, since there is no provision for "regular" users to access EXE directly (as described above), and the portion of EXE they access and control are outside the scope of the TOE.

Primary management is done using the 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 of these services can be managed from the web browser, which uses a menu-driven navigation system.

There is also a very simple serial-based connection (RS-232) that provides a simple menu interface. This is used to configure the IP interface (IP address, etc.). It is password-protected.

#### 1.4.3.5 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 is 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.

#### 1.4.3.6 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.

#### 1.4.3.7 Trusted Path/Channels

The TOE allows the establishment of a trusted channel between a video control system (such as Evertz' Magnum) and the EXE. The TOE also establishes a secure connection for sending syslog data to a syslog server using TLS. The TOE also provides a trusted path to Security Administrators via HTTPS/TLS.

#### 1.4.3.8 Excluded Functionality

The TOE includes the following functionality that may not be enabled or used in in the CC evaluated configuration:

- SNMP Traps

### 1.4.4 TOE Documentation

Evertz Microsystems, Ltd. publishes manuals detailing the installation configuration and operation of the EXE and Magnum control software. These are available to customers both on paper and as electronic copies. Electronic copies are available in .pdf format from the Evertz support website with a valid customer login.

| CATEGORY | PRODUCT | MANUAL |
|---|---|---|
| **TOE MODELS** | MMA10G-EXE16 | MMA10G-EXE Series-CC High Bandwidth 10GE Switch Fabric User Manual, Version 1.0, January 2017<br>[EXE UG] |
| | MMA10G-EXE26 | |
| | MMA10G-EXE36 | |
| | EXE2.0-16-10G-A1 | |
| | EXE2.0-16-25G-A1 | |
| | EXE2.0-26-10G-A1 | |
| | EXE2.0-26-25G-A1 | |
| | EXE2.0-36-10G-A1 | |
| | EXE2.0-36-25G-A1 | |
| | MMA10G-IPX-128 | MMA10G-EXE Series-CC High Bandwidth 10GE Switch Fabric User Manual, Version 1.0, January 2017<br>[IPX128 UG] |

**Table 6 Evertz Operating Manuals**

In addition, the following Common Criteria documentation is included:

- MMA10G-EXE Security Target v1.1, May 5, 2020
- MMA10G-EXE Guidance Documentation v5.6, May 1, 2020

### 1.4.5 Other References

- collaborative Protection Profile for Network Devices, Version 2.1 [NDcPP]

# 2 Conformance Claims

## 2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017: Part 3 extended

## 2.2 Protection Profile Conformance

This TOE is conformant to:

- collaborative Protection Profile for Network Devices, Version 2.1 [NDcPP]

## 2.3 Conformance Rationale

This Security Target provides exact conformance to the collaborative Protection Profile for Network Devices, Version 2.1 [NDcPP]. The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile performing only operations defined there.

### 2.3.1 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date that are applicable to [NDcPP] have been addressed. The following table identifies all applicable TD:

| Identifier | Applicable | Exclusion Rationale (if applicable) |
|---|---|---|
| 0484 – NIT Technical Decision for Interactive sessions in FTA_SSL_EXT.1 & FTA_SSL.3 | Yes | |
| 0483 – NIT Technical Decision for Applicability of FPT_APW_EXT.1 | Yes | |
| 0482 – NIT Technical Decision for Identification of usage of cryptographic schemes | Yes | |
| 0481 – NIT Technical Decision for FCS_(D)TLSC_EXT.X.2 IP addresses in reference identifiers | Yes | |
| 0480 – NIT Technical Decision for Granularity of audit events | Yes | |
| 0478 – NIT Technical Decision for Application Notes for FIA_X509_EXT.1 iterations | Yes | |
| 0477 – NIT Technical Decision for Clarifying FPT_TUD_EXT.1 Trusted Update | Yes | |
| 0475 – NIT Technical Decision for Separate traffic consideration for SSH rekey | No | FCS_SSHS/SSHC_EXT.1 is not claimed. |
| 0453 – NIT Technical Decision for Clarify authentication methods SSH clients can use to authenticate SSH | No | FCS_SSHC_EXT is not claimed. |
| 0451 – NIT Technical Decision for ITT Comm UUID Reference Identifier | Yes | |
| 0450 – NIT Technical Decision for RSA-based ciphers and the Server Key Exchange message | Yes | |
| 0447 – NIT Technical Decision for Using 'diffie'hellman-group-exchange-sha256' in FCS_SSHC/S_EXT.1.7 | No | FCS_SSH*_EXT.1 is not claimed. |

| Identifier | Applicable | Exclusion Rationale (if applicable) |
|---|---|---|
| 0425 – NIT Technical Decision for Cut-and-paste Error for Guidance AA | Yes | |
| 0424 – NIT Technical Decision for NDcPP v2.1 Clarification – FCS_SSHC/S_EXT.1.5 | No | SSH functionality not included in the evaluation. |
| 0423 – NIT Technical Decision for Clarification about application of RFI#201726rev2 | Yes | |
| 0412 – NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy | No | FCS_SSHS_EXT is not claimed. |
| 0411 – NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 – Server and client side seem to be confused | No | FCS_SSHC_EXT is not claimed. |
| 0410 – NIT Technical Decision for Redundant assurance activities associated with FAU_GEN.1 | Yes | |
| 0409 – NIT Technical Decision for Applicability of FIA_AFL.1 to key-based SSH authentication | No | SSH functionality is not claimed. |
| 0408 – NIT Technical Decision for local vs. remote administrator accounts | Yes | |
| 0407 – NIT Technical Decision for handling Certification of Cloud Deployments | No | TOE is not cloud-based. |
| 0402 – NIT Technical Decision for RSA-based FCS_CKM.2 Selection | Yes | |
| 0401 – NIT Technical Decision for Reliance on external severs to meet SFRs | Yes | |
| 0400 – NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment | Yes | |
| 0399 – NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2) | Yes | |
| 0398 – NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR | No | SSH is not included in the evaluation. |
| 0397 – NIT Technical Decision for Fixing AES-CTR Mode Tests | No | AES-CTR mode is not used in the TOE. |
| 0396 – NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2 | Yes | |
| 0395 – NIT Technical Decision for Different Handling of TLS 1.1 and TLS 1.2 | Yes | |

**Table 7 Technical Decisions**

# 3 Security Problem Definition

The security problem definition has been taken from [NDcPP] and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any organizational security policies that the TOE is expected to enforce.

## 3.1 Threats

The following threats are drawn directly from the [NDcPP].

| ID | Threat |
|---|---|
| T.UNAUTHORIZED_ADMINISTRATOR_ACCESS | Threat agents may attempt to gain administrator access to the network device by nefarious means such as masquerading as an administrator to the device, masquerading as the device to an administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between network devices. Successfully gaining administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides. |
| T.WEAK_CRYPTOGRAPHY | Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort. |
| T.UNTRUSTED_COMMUNICATION_CHANNELS | Threat agents may attempt to target network devices that do not use standardized secure tunneling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the network device itself. |
| T.WEAK_AUTHENTICATION_ENDPOINTS | Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised. |
| T.UPDATE_COMPROMISE | Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration. |

| | |
|---|---|
| T.UNDETECTED_ACTIVITY | Threat agents may attempt to access, change, and/or modify the security functionality of the network device without administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the administrator would have no knowledge that the device has been compromised. |
| T.SECURITY_FUNCTIONALITY_COMPROMISE | Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker. |
| T.PASSWORD_CRACKING | Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices. |
| T.SECURITY_FUNCTIONALITY_FAILURE | An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device. |

**Table 8 Threats**

## 3.2 Assumptions

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

| ID | Assumption |
|---|---|
| A.PHYSICAL_PROTECTION | The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. |
| A.LIMITED_FUNCTIONALITY | The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide computing platform for general purpose applications (unrelated to networking functionality). |
| A.NO_THRU_TRAFFIC_PROTECTION | A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of network devices (e.g, firewall). |

| | |
|---|---|
| A.TRUSTED_ADMINISTRATOR | The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious administrator that actively works to bypass or compromise the security of the device.<br>For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trusted source (aka 'root store', 'trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification). |
| A.REGULAR_UPDATES | The network device firmware and software is assumed to be updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities. |
| A.ADMIN_CREDENTIALS_SECURE | The administrator's credentials (private key) used to access the network device are protected by the platform on which they reside. |
| A.COMPONENTS_RUNNING (applies to distributed TOEs only) | For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components. |
| A.RESIDUAL_INFORMATION | The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. |

**Table 9 Assumptions**

## 3.3 Organizational Security Policies

The following Organizational Security Policies are drawn directly from the [NDcPP].

| ID | OSP |
|---|---|
| P.ACCESS_BANNER | The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE. |

**Table 10 OSPs**

# 4 Security Objectives

The security objectives have been taken from [NDcPP] and are reproduced here for the convenience of the reader.

## 4.1 Security Objectives for the Operational Environment

The following security objectives for the operational environment assist the TOE in correctly providing its security functionality. These track with the assumptions about the environment.

| ID | Objective for the Operation Environment |
|---|---|
| OE.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment. |
| OE.NO_GENERAL_PURPOSE | There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. |
| OE.NO_THRU_TRAFFIC_PROTECTION | The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment. |
| OE.TRUSTED_ADMIN | Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.<br>For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted. |
| OE.UPDATES | The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities. |
| OE.ADMIN_CREDENTIALS_SECURE | The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside. |
| OE.COMPONENTS_RUNNING<br>(applies to distributed TOEs only) | For distributed TOEs the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly. |
| OE.RESIDUAL_INFORMATION | The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. |

**Table 11 Objectives for the Operational Environment**

# 5 Security Requirements

This section identifies the Security Functional Requirements for the TOE. The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, dated: April 2017 and all international interpretations.

| Requirement | Description |
|---|---|
| FAU_GEN.1 | Audit data generation |
| FAU_GEN.2 | User identity association |
| FAU_STG_EXT.1 | Protected Audit Event Storage |
| FCS_CKM.1 | Cryptographic Key Generation |
| FCS_CKM.2 | Cryptographic Key Establishment |
| FCS_CKM.4 | Cryptographic Key Destruction |
| FCS_COP.1/DataEncryption | Cryptographic Operation (AES Data Encryption/Decryption) |
| FCS_COP.1/SigGen | Cryptographic Operation (Signature Generation and Verification) |
| FCS_COP.1/Hash | Cryptographic Operation (Hash Algorithm) |
| FCS_COP.1/KeyedHash | Cryptographic Operation (Keyed Hash Algorithm) |
| FCS_HTTPS_EXT.1 | HTTPS Protocol |
| FCS_RBG_EXT.1 | Random Bit Generation |
| FCS_TLSC_EXT.1 | TLS Client Protocol |
| FCS_TLSS_EXT.1 | TLS Server Protocol |
| FIA_AFL.1 | Authentication Failure Management |
| FIA_PMG_EXT.1 | Password Management |
| FIA_UIA_EXT.1 | User Identification and Authentication |
| FIA_UAU_EXT.2 | Password-based Authentication Mechanism |
| FIA_UAU.7 | Protected Authentication Feedback |
| FIA_X509_EXT.1/Rev | X.509 Certificate Validation |
| FIA_X509_EXT.2 | Certificate Authentication |
| FIA_X509_EXT.3 | Certificate Requests |
| FMT_MOF.1/Functions | Management of security functions behavior |
| FMT_MOF.1/ManualUpdate | Management of security functions behaviour |
| FMT_MTD.1/CoreData | Management of TSF Data |
| FMT_MTD.1/CryptoKeys | Management of TSF data |
| FMT_SMF.1 | Specification of Management Functions |
| FMT_SMR.2 | Restrictions on security roles |
| FPT_SKP_EXT.1 | Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) |
| FPT_APW_EXT.1 | Protection of Administrator Passwords |
| FPT_TST_EXT.1 | TSF Testing |
| FPT_TUD_EXT.1 | Trusted Update |
| FPT_STM_EXT.1 | Reliable Time Stamps |
| FTA_SSL_EXT.1 | TSF-initiated Session Locking |
| FTA_SSL.3 | TSF-initiated Termination |
| FTA_SSL.4 | User-initiated Termination |
| FTA_TAB.1 | Default TOE Access Banner |
| FTP_ITC.1 | Inter-TSF trusted channel |

| | |
|---|---|
| FTP_TRP.1/Admin | Trusted Path |

**Table 12 SFRs**

## 5.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document follows the conventions used in NDcPP v2.1 in order to comply with exact conformance. Within selections and assignments made in the ST the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with [*italicized*] text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with [underlined] text;
- Selection within a selection: Indicated by an additional set of [brackets];
- Iteration: Indicated by appending the iteration identifier after a slash, e.g., /SigGen.
- Where operations were completed in the PP itself, the formatting used in the PP has been retained.

Extended SFRs are identified by having a label 'EXT' after the requirement name. Formatting conventions outside of operations matches the formatting specified within the PP.

## 5.2 Security Functional requirements

### 5.2.1 Security Audit (FAU)

#### 5.2.1.1 FAU_GEN.1 Audit data generation

**FAU_GEN.1.1**

The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) *All administrative actions comprising:*

- *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
- *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
- *Resetting passwords (name of related user account shall be logged).*
- [[*no other actions*]]*;*

d) *Specifically defined auditable events listed in Table 13*.

**FAU_GEN.1.2**

The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of Table 13*.

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_GEN.1 | None. | None. |
| FAU_GEN.2 | None. | None. |
| FAU_STG_EXT.1 | None. | None. |
| FCS_CKM.1 | None. | None. |
| FCS_CKM.2 | None. | None. |
| FCS_CKM.4 | None. | None. |
| FCS_COP.1/DataEncryption | None. | None. |
| FCS_COP.1/SigGen | None. | None. |
| FCS_COP.1/Hash | None. | None. |
| FCS_COP.1/KeyedHash | None. | None. |
| FCS_HTTPS_EXT.1 | Failure to establish a HTTPS Session. | Reason for failure |
| FCS_RBG_EXT.1 | None. | None. |
| FCS_TLSC_EXT.1 | Failure to establish a TLS Session | Reason for failure |
| FCS_TLSS_EXT.1 | Failure to establish a TLS Session | Reason for failure |
| FIA_AFL.1 | Unsuccessful login attempts limit is met or exceeded. | Origin of the attempt (e.g., IP address). |
| FIA_PMG_EXT.1 | None. | None. |
| FIA_UIA_EXT.1 | All use of identification and authentication mechanism. | Origin of the attempt (e.g., IP address). |
| FIA_UAU_EXT.2 | All use of identification and authentication mechanism. | Origin of the attempt (e.g., IP address). |
| FIA_UAU.7 | None. | None. |
| FIA_X509_EXT.1/Rev | Unsuccessful attempt to validate a certificate<br>Any addition, replacement or removal of trust anchors in the TOE's trust store. | Reason for failure of certificate validation<br>Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store |
| FIA_X509_EXT.2 | None | None |
| FIA_X509_EXT.3 | None. | None. |
| FMT_MOF.1/Functions | None. | None. |
| FMT_MOF.1/ManualUpdate | Any attempt to initiate a manual update | None. |
| FMT_MTD.1/CoreData | None. | None. |
| FMT_MTD.1/CryptoKeys | Management of cryptographic keys. | None. |
| FMT_SMF.1 | All management activities of TSF data. | None. |
| FMT_SMR.2 | None. | None. |
| FPT_SKP_EXT.1 | None. | None. |
| FPT_APW_EXT.1 | None. | None. |
| FPT_TST_EXT.1 | None. | None. |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FPT_TUD_EXT.1 | Initiation of update; result of the update attempt (success or failure) | None. |
| FPT_STM_EXT.1 | Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1) | For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address). |
| FTA_SSL_EXT.1 (if "terminate the session" is selected) | Any attempts at unlocking of an interactive session. | None. |
| FTA_SSL.3 | The termination of a remote session by the session locking mechanism. | None. |
| FTA_SSL.4 | The termination of an interactive session. | None. |
| FTA_TAB.1 | None. | None. |
| FTP_ITC.1 | Initiation of the trusted channel.<br>Termination of the trusted channel.<br>Failure of the trusted channel functions. | Identification of the initiator and target of failed trusted channels establishment attempt. |
| FTP_TRP.1/Admin | Initiation of the trusted path.<br>Termination of the trusted path.<br>Failure of the trusted path functions. | None. |

**Table 13 Security Functional Requirements and Auditable Events**

### 5.2.1.2 FAU_GEN.2 User identity association

**FAU_GEN.2.1**

For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

### 5.2.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

**FAU_STG_EXT.1.1**

The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

**FAU_STG_EXT.1.2**

The TSF shall be able to store generated audit data on the TOE itself. [

- TOE shall consist of a single standalone component that stores audit data locally.]

**FAU_STG_EXT.1.3**

The TSF shall [overwrite previous audit records according to the following rule: [*on a circular (FIFO) basis*]] when the local storage space for audit data is full.

## 5.2.2 Cryptographic Support (FCS)

### 5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

**FCS_CKM.1.1**

The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- 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;
- ECC schemes using “NIST curves” [P-256, P-384] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4;

~~*]* and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*]~~.

### 5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment

**FCS_CKM.2.1**

The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1”
- Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 2, “Recommended for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”

~~*]* that meets the following: [assignment: *list of standards*]~~.

***ST Application Note***
*FCS_CKM.2 was updated based on TD0402.*

### 5.2.2.3 FCS_CKM.4 Cryptographic Key Destruction

**FCS_CKM.4.1**

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [

- *For plaintext keys in volatile storage, the destruction shall be executed by a* [single overwrite consisting of [zeroes]]*;*
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that:* [
    - logically addresses the storage location of the key and performs a [[single] overwrite consisting of [zeros]]*;*

that meets the following: *No Standard*.]

### 5.2.2.4 FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)

**FCS_COP.1.1/DataEncryption**

The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic *algorithm AES used in* [CBC, GCM] *mode* and cryptographic key sizes [128 bits, 256 bits] that meet the following: *AES as specified in ISO 18033-3*, [CBC as specified in ISO 10116, GCM as specified in ISO 19772].

### 5.2.2.5 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

**FCS_COP.1.1/SigGen**

The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- *RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits, 3072 bits]*

]

that meet the following: [

- *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,*]

### 5.2.2.6 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

**FCS_COP.1.1/Hash**

The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [*SHA-1, SHA-256, SHA-384*] ~~and cryptographic key sizes [*assignment: cryptographic key sizes*~~] and **message digest sizes [*160, 256*, 384] bits** that meet the following: *ISO/IEC 10118-3:2004*.

### 5.2.2.7 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

**FCS_COP.1.1/KeyedHash**

The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [*HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384*] and cryptographic key sizes *[160, 256, 384]* **and message digest sizes [*160, 256, 384*] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

### 5.2.2.8 FCS_HTTPS_EXT.1 HTTPS Protocol

**FCS_HTTPS_EXT.1.1**

The TSF shall implement the HTTPS protocol that complies with RFC 2818.

**FCS_HTTPS_EXT.1.2**

The TSF shall implement HTTPS using TLS.

**FCS_HTTPS_EXT.1.3**

If a peer certificate is presented, the TSF shall [*not establish the connection*] if the peer certificate is deemed invalid.

### 5.2.2.9 FCS_RBG_EXT.1 Random Bit Generation

**FCS_RBG_EXT.1.1**

The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [*CTR_DRBG (AES)*].

**FCS_RBG_EXT.1.2**

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*two*] software-based noise sources] with a minimum of [*256 bits*] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

### 5.2.2.10 FCS_TLSC_EXT.1 TLS Client Protocol

**FCS_TLSC_EXT.1.1**

The TSF shall implement [*TLS 1.2 (RFC 5246)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- *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*
  *].*

**FCS_TLSC_EXT.1.2**

The TSF shall verify that the presented identifiers of the following types: [identifiers defined in RFC 6125] are matched to reference identifiers.

***ST Application Note***
*FCS_TLSC_EXT.1.2 was updated based on TD0481.*

**FCS_TLSC_EXT.1.3**

When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [

- *Not implement any administrator override mechanism*
  ].

**FCS_TLSC_EXT.1.4**

The TSF shall [*present the Supported Elliptic Curves Extension with the following NIST curves:* [*secp256r1, secp384r1*] *and no other curves*] in the Client Hello.

### 5.2.2.11 FCS_TLSS_EXT.1 TLS Server Protocol

*Note: This SFR covers communication between administrators and the TOE.*

**FCS_TLSS_EXT.1.1**

The TSF shall implement [*TLS 1.2 (RFC 5246)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- *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*
  ].

**FCS_TLSS_EXT.1.2**

The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [*TLSv1.1*].

**FCS_TLSS_EXT.1.3**

The TSF shall [*perform RSA key establishment with key size* [2048 bits]*; generate EC Diffie-Hellman parameters over NIST curves* [secp256r1, secp384r1] *and no other curves*].

## 5.2.3 Identification and Authentication (FIA)

### 5.2.3.1 FIA_AFL.1 Authentication Failure Management

**FIA_AFL.1.1**

The TSF shall detect when an Administrator configurable positive integer within [*3 to 20*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

**FIA_AFL.1.2**

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [*prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until [an Administrator unlocks the user] is taken by an Administrator]*.

***ST Application Note***
*FIA_AFL.1 was updated based on TD0408.*

### 5.2.3.2 FIA_PMG_EXT.1 Password Management

**FIA_PMG_EXT.1.1**

The TSF shall provide the following password management capabilities for administrative passwords:

*a)* Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [*“!”; “@”; “#”; “$”; “%”; “^”; “&”; “*”; “(“; “)”;*];
*b)* Minimum password length shall be configurable to between [*15*] and [*20*] characters.

### 5.2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

**FIA_UIA_EXT.1.1**

The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [*No other actions*].

**FIA_UIA_EXT.1.2**

The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

### 5.2.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

**FIA_UAU_EXT.2.1**

The TSF shall provide a local [*password-based]* authentication mechanism to perform local administrative user authentication.

***ST Application Note***
*FIA_UAU_EXT.2.1 was updated based on TD0408.*

### 5.2.3.5 FIA_UAU.7 Protected Authentication Feedback

**FIA_UAU.7.1**

The TSF shall provide only *obscured feedback* to the administrative user while the authentication is in progress **at the local console**.

### 5.2.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

**FIA_X509_EXT.1.1/Rev**

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation **supporting a minimum path length of three certificates**.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
  - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

**FIA_X509_EXT.1.2/Rev**

The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

#### 5.2.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

**FIA_X509_EXT.2.1**

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [*HTTPS, TLS*], and [*no additional uses*].

**FIA_X509_EXT.2.2**

When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [*accept the certificate*].

#### 5.2.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

**FIA_X509_EXT.3.1**

The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [Common Name, Organization, Organizational Unit, Country].

**FIA_X509_EXT.3.2**

The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

### 5.2.4 Security Management (FMT)

#### 5.2.4.1 FMT_MOF.1/Functions Management of security functions behavior

**FMT_MOF.1/Functions**

The TSF shall restrict the ability to [modify the behavior of] the functions [*transmission of audit data to an external IT entity*] to *Security Administrators.*

#### 5.2.4.2 FMT_MOF.1/ManualUpdate Management of security functions behaviour

**FMT_MOF.1.1/ManualUpdate**

The TSF shall restrict the ability to enable the functions *to perform manual updates* to *Security Administrators*.

#### 5.2.4.3 FMT_MTD.1/CoreData Management of TSF Data

**FMT_MTD.1.1/CoreData**

The TSF shall restrict the ability to *manage* the *TSF data* to *Security Administrators*.

#### 5.2.4.4 FMT_MTD.1/CryptoKeys Management of TSF Data

**FMT_MTD.1.1/CryptoKeys**

The TSF shall restrict the ability to *manage* the *cryptographic keys* to *Security Administrators*.

#### 5.2.4.5 FMT_SMF.1 Specification of Management Functions

**FMT_SMF.1.1**

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 behaviour;*
    - *Ability to manage cryptographic keys;*
    - *Ability to set the time which is used for time-stamps;*
    - *Ability to re-enable an Administrator account;*
    - *Ability to import X.509v3 certificates to the TOE's trust store*].

#### 5.2.4.6 FMT_SMR.2 Restrictions on security roles

**FMT_SMR.2.1**

The TSF shall maintain the roles:

- *Security Administrator.*

**FMT_SMR.2.2**

The TSF shall be able to associate users with roles.

**FMT_SMR.2.3**

The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely*

are satisfied.

### 5.2.5 Protection of the TSF (FPT)

#### 5.2.5.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

**FPT_SKP_EXT.1.1**

The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

#### 5.2.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

**FPT_APW_EXT.1.1**

The TSF shall store administrative passwords in non-plaintext form.

**FPT_APW_EXT.1.2**

The TSF shall prevent the reading of plaintext administrative passwords.

***ST Application Note***
*FPT_APW_EXT.1 was updated based on TD0483.*

### 5.2.5.3 FPT_TST_EXT.1 TSF Testing

**FPT_TST_EXT.1.1**

The TSF shall run a suite of the following self-tests [*during initial start-up (on power on)*] to demonstrate the correct operation of the TSF: [

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

].

### 5.2.5.4 FPT_TUD_EXT.1 Trusted Update

**FPT_TUD_EXT.1.1**

The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [*the most recently installed version of the TOE firmware/software*].

**FPT_TUD_EXT.1.2**

The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [*no other update mechanism*].

**FPT_TUD_EXT.1.3**

The TSF shall provide means to authenticate firmware/software updates to the TOE using a [*digital signature mechanism*] prior to installing those updates.

### 5.2.5.5 FPT_STM_EXT.1 Reliable Time Stamps

**FPT_STM_EXT.1.1**

The TSF shall be able to provide reliable time stamps for its own use.

**FPT_STM_EXT.1.2**

The TSF shall [*allow the Security Administrator to set the time*].

## 5.2.6 TOE Access (FTA)

### 5.2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

**FTA_SSL_EXT.1.1**

The TSF shall, for local interactive sessions, [

- *terminate the session*]

after a Security Administrator-specified time period of inactivity.

### 5.2.6.2 FTA_SSL.3 TSF-initiated Termination

**FTA_SSL.3.1**

The TSF shall terminate **a remote** interactive session after a *Security Administrator-configurable time interval of session inactivity*.

### 5.2.6.3 FTA_SSL.4 User-initiated Termination

**FTA_SSL.4.1**

The TSF shall allow **Administrator**-initiated termination of the **Administrator's** own interactive session.

### 5.2.6.4 FTA_TAB.1 Default TOE Access Banners

**FTA_TAB.1.1**

Before establishing **an administrative user** session the TSF shall display **a Security Administrator-specified** advisory **notice and consent** warning message regarding use of the TOE.

## 5.2.7 Trusted path/channels (FTP)

### 5.2.7.1 FTP_ITC.1 Inter-TSF trusted channel

**FTP_ITC.1.1**

The TSF shall **be capable of using [*TLS*] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [*video switch control system (such as Evertz MAGNUM)*]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **disclosure and detection of modification of the channel data**.

**FTP_ITC.1.2**

The TSF shall permit **the TSF or the authorized IT entities** to initiate communication via the trusted channel.

**FTP_ITC.1.3**

The TSF shall initiate communication via the trusted channel for [*auditing services and system logging, controlling video switches*].

### 5.2.7.2 FTP_TRP.1/Admin Trusted Path

**FTP_TRP.1.1/Admin**

The TSF shall **be capable of using [*TLS, HTTPS*] to** provide a communication path between itself and **authorized** remote **Administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **disclosure and provides detection of modification of the channel data.**

**FTP_TRP.1.2/Admin**

The TSF shall permit remote **Administrators** to initiate communication via the trusted path.

**FTP_TRP.1.3/Admin**

The TSF shall require the use of the trusted path for *initial Administrator authentication and all remote administration actions*.

## 5.3 TOE SFR Dependencies Rationale for SFRs

[NDcPP] contains all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP has been approved.

## 5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from [NDcPP] which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

| Assurance Class | Components | Components Description |
|---|---|---|
| Development | ADV_FSP.1 | Basic Functional Specification |
| Guidance Documents | AGD_OPE.1 | Operational User Guidance |
| | AGD_PRE.1 | Preparative User Guidance |
| Life Cycle Support | ALC_CMC.1 | Labeling of the TOE |
| | ALC_CMS.1 | TOE CM Coverage |
| Tests | ATE_IND.1 | Independent Testing – Conformance |
| Vulnerability Assessment | AVA_VAN.1 | Vulnerability Analysis |

**Table 14 Security Assurance Requirements**

## 5.5 Rationale for Security Assurance Requirements

The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.

## 5.6 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Evertz to satisfy the assurance requirements. The table below lists the details.

| SAR Component | How the SAR will be met |
|---|---|
| ADV_FSP.1 | The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). |
| AGD_OPE.1 | The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance. |
| AGD_PRE.1 | The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration. |
| ALC_CMC.1 | The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated. |
| ALC_CMS.1 | |
| ATE_IND.1 | Evertz Microsystems will provide the TOE for testing. |
| AVA_VAN.1 | Evertz Microsystems will provide the TOE for testing.<br>Evertz Microsystems will provide a document identifying the list of software and hardware components. |

**Table 15 TOE Security Assurance Measures**

# 6 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

| TOE SFR | Rationale |
|---|---|
| FAU_GEN.1<br>FAU_GEN.2 | EXE generates audit logs that consist of various auditable events or actions. This includes logins, use of trusted channel/path and cryptographic operations. Each audit event contains an associated date/time stamp, a label for the type of event, a user ID (if applicable), a description of the event and any additional information specified in column three of Table 13.<br>Audit records are created when an auditable event that belongs to a set of predefined events had occurred. The set of auditable events can be sub-categorized into functional events and access events. The TSF generates audit records for the following events:<br>• Startup and shutdown of the audit function<br>• Administrative login and logout events<br>• Changes to TSF data related to configuration changes<br>• Generation of a CSR and associated keypair<br>• Installation, removal or replacement of a certificate<br>• Resetting passwords<br>• Failure to establish a HTTPS/TLS session<br>• Failure to establish a TLS session<br>• All use of the identification and authentication mechanism (local and remote connections to the TSF)<br>• Unsuccessful attempts to validate a certificate<br>• Initiation of a software update<br>• Result of a software update<br>• Changes to the time<br>• Modification of the behavior of the TSF<br>• Failure of self-tests<br>• Initiation and termination of the trusted channel<br>• Initiation and termination of the trusted path<br>• Attempts to unlock an interactive session<br>• Termination of a session by the session locking mechanism<br>There is only one server key stored on the TOE. Any action that is logged referring to this key identifies the key as the server certificate. |
| FAU_STG_EXT.1 | The TOE is a standalone TOE. 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 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. 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.<br>The TOE overwrites previous audit records on a circular (FIFO) basis when the local 1.2G (for /var/log) and 1G (for /nv/syslog/current) storage space for audit is full.<br>The TSF implements Syslog over TLS using TLS v1.2, acting as a syslog client. Logs are sent to the Syslog servers in real-time. The trusted channel with the Syslog server is described in greater detail in the FCS_TLSC_EXT.1 description. |

<table>
<tr><th>TOE SFR</th><th>Rationale</th></tr>
<tr><td>FCS_CKM.1</td><td>The TSF generates RSA keys of at least 2048 bits in length for digital signatures and key establishment in support of TLS sessions. The TSF generates ECDSA keys using NIST curves P-256 or P-384 to generate EC-DHE components for key establishment in TLS sessions.<br>The CAVP algorithm certificates for the TOE are included in Table 5.</td></tr>
<tr><td>FCS_CKM.2</td><td>The TOE performs key establishment using Elliptic Curve Diffie Hellman (EC DHE) ephemeral keys with P-256 and P-384 curves, and RSA with 2048-bit keypairs. The below table describe the key establishment schemes and where they are used.
<table>
<tr><td>Scheme</td><td>SFR</td><td>Service</td></tr>
<tr><td rowspan="3">RSA</td><td>FCS_TLSS_EXT.2</td><td>MAGNUM</td></tr>
<tr><td>FCS_TLSS_EXT.1</td><td>Administration</td></tr>
<tr><td>FCS_TLSC_EXT.1</td><td>Audit Server</td></tr>
<tr><td rowspan="3">ECDH</td><td>FCS_TLSS_EXT.2</td><td>MAGNUM</td></tr>
<tr><td>FCS_TLSS_EXT.1</td><td>Administration</td></tr>
<tr><td>FCS_TLSC_EXT.1</td><td>Audit Server</td></tr>
</table>
The CAVP algorithm certificates for the TOE are included in Table 5.</td></tr>
<tr><td>FCS_CKM.4</td><td>Cryptographic keys are destroyed by first overwriting the key file content with all zeros. This will be done three times. Then a read-verification will be performed to ensure that the entire content has really been changed to zeros and not any other values. If this step fails, the above process is repeated one more time. If this step fails again, a warning is displayed to the administrator. All persistently stored keys are conformant with this process.<br>The interface used to destroy cryptographic keys is via the <b>zeroize</b> command at the local serial console.<br>The following keys are stored:
<ul>
<li>the server certificate, which is generated by the TOE and used for HTTP web service and Magnum TLS connection;</li>
<li>the private key matching the server certificate, which is generated by the TOE and used for de-encryption;</li>
</ul>
All of these cryptographic keys are stored in plaintext on nonvolatile NOR flash storage. No direct interface/access is provided to view or modify the contents of these files.<br>Ephemeral TLS keys are stored in volatile memory and are destroyed automatically when the TLS session is closed.</td></tr>
<tr><td>FCS_COP.1/DataEncryption</td><td>The TOE provides AES encryption/decryption in CBC and GCM mode with 128- and 256-bit keys.<br>The CAVP algorithm certificates for the TOE are included in Table 5.</td></tr>
<tr><td>FCS_COP.1/SigGen</td><td>The TOE supports signature generation and verification with RSA (2048- and, 3072-bit) with SHA-256/384 in accordance with FIPS PUB 186-4.</td></tr>
</table>

| TOE SFR | Rationale |
|---|---|
| | These signatures are used in support of TLS authentication.<br>The CAVP algorithm certificates for the TOE are included in Table 5. |
| FCS_COP.1/Hash | 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.<br>The CAVP algorithm certificates for the TOE are included in Table 5. |
| FCS_COP.1/KeyedHash | Keyed-hash message authentication is used as part of TLS protocol as part of the negotiated cipher suites between peers and as part of the firmware integrity check on upgrades.<br>HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA384 are the keyed-hash message authentication functions used by EXE. The following parameters are supported for HMAC functions within the TOE:<br>• Hash: SHA-1, SHA-256, SHA-384<br>• Key Length: 160, 256, 384 bits<br>• Block size: 512 bits (SHA-1 and SHA-256), 1024 bits (SHA-384)<br>• MAC length: 160, 256, 384 bits<br>The CAVP algorithm certificates for the TOE are included in Table 5. |
| FCS_HTTPS_EXT.1<br>FCS_TLSS_EXT.1<br>FCS_TLSC_EXT.1 | Remote management of the TOE is performed via a web GUI accessible over an RFC2818 conformant HTTPS session protected with TLSv1.2. All other versions of TLS and/or SSL are rejected.<br>TLS v1.2 is used to export audit records, communicate with a video switch control system, and support remote administrator connections. Server-side and client-side TLS work the same. EXE acts as the server in connections with remote administrators and with the MAGNUM video switch controller.<br>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:<br>• TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268<br>• TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268<br>• TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246<br>• TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246<br>• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289<br>• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289<br>EXE supports cipher suites that use ECDHE and RSA schemes for key exchange and RSA keys for authentication. The ECDHE server key exchange message conforms to RFC 5246 section 7.4.3. These keys are generated with OpenSSL's command line internally to the TSF. By default, the TOE supports the Elliptic Curves Extension with NIST elliptic curves secp256r1 and secp384r1. The RSA key establishment scheme uses 2048 bit keys.<br>When validating a server's certificate, EXE uses CRL (certification revocation list) to check for invalid certificates. The TOE pulls the CRL file from the CRL-DP to use by |

| TOE SFR | Rationale |
|---|---|
|  | EXE during certificate validation process to check for revocation status of the peer certificates.<br>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. EXE supports FQDN identifier types only. SRV-ID and URI-ID types are not supported.<br>EXE does not support certificate pinning.<br>EXE supports wildcard in certificates. The wildcard must be in the left-most label of the presented identifier and can only covers 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. |
| FCS_RBG_EXT.1 | 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.<br>The entropy sources are discussed in greater detail in the Entropy documentation.<br>The CAVP algorithm certificates for the TOE are included in Table 5. |
| FIA_AFL.1 | 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.<br>If the user enters an incorrect password the configured number of times, the user is locked out they cannot login through any remote interface on the TOE. Users must have an administrator unlock their account before they can regain access. Administrators can also have a different administrator unlock their account.<br>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. |
| FIA_PMG_EXT.1 | EXE enforces that passwords must meet minimum requirements such as length, mix of number, lower/upper case letters, and the following special characters “!”; “@”; “#”; “$”; “%”; “^”; “&”; “*”; “(“; “)”;. No common dictionary words are allowed. Passwords must be at least a minimum length settable by the administrator and support 15 to 20 characters. |
| FIA_UIA_EXT.1<br>FIA_UAU_EXT.2 | The only accounts that the EXE will establish are Security Administrator accounts. Users only control the EXE indirectly via MAGNUM. CO/Administrators are identified and authenticated via username and password prior to performing any operations other than acknowledging the warning banner. The EXE CO/Administrators user accounts module maintains Security Administrator credentials. Since the only role that accesses the EXE directly is that of Security Administrator there is no assignment of roles required.<br>Administrators can logon via the web interface using HTTPS or locally on the serial port.   A username and password is to authenticate the administrator for both methods. The Security Administrator is considered authenticated if the username and password match the stored credential values. |

| TOE SFR | Rationale |
|---|---|
| | 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. |
| FIA_UAU.7 | When the user is entering their password over the local console, the TSF does not echo any characters back. |
| FIA_X509_EXT.1/Rev | 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. 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.<br>If certificates are uploaded to EXE for its own use those certificates are checked upon upload. The full certificate chain presented by TLS servers are validated during the establishment of a TLS connection.<br>For an expired certificate, EXE will deny the connection. EXE also uses CRL to verify whether the 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.<br>The TSF verifies the validity of a certificate when:<br>• An HTTPS client establishes a TLS connection (HTTPS Server Certificate)<br>If the Security Administrator loads a certificate with a Subject Type=CA, the TSF does not validate the certificate path. |
| FIA_X509_EXT.2 | Instructions about generating/downloading CSR and loading certificate can be found on EXE manual. The Administrator can only upload one certificate and one 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. If certificate verification fails due to failure to establish a connection to the CRL distribution point, the connection attempt is permitted, however a log is generated indicating the reason for validation failure. |
| FIA_X509_EXT.3 | The TSF allows Security Administrators to generate Certificate Signing Requests. The TSF requires the Security Administrator to specify the following values:<br>• Common Name<br>• Organization<br>• Locality<br>• State<br>• Country<br>• Key Length (2048)<br>A CSR can be generated from the serial console menu. When validating certificates, each certificate from the chain is sequentially validated, terminating at the root CA. If any invalid certificate is found in this process, the validation fails. |
| FMT_MOF.1/Functions<br>FMT_MOF.1/ManualUpdate<br>FMT_MTD.1/CoreData<br>FMT_MTD.1/CryptoKeys<br>FMT_SMF.1<br>FMT_SMR.2 | EXE gives the Security Administrator the ability to manage the security functions: auditing operations, administrative user accounts, password and session policies, advisory banners, software updates, as well as cryptographic functions. EXE ensures that only secure values are accepted for security attributes. A Security Administrator can change passwords, and can add, edit and/or delete Security Administrator accounts. The (non-administrative) User has no direct access or |

| TOE SFR | Rationale |
|---|---|
| | control over EXE; a (non-administrative) User may only access an EXE card through MAGNUM. No administrative functionality is available prior to login.<br>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. The web interface and local console allow the Security Administrator to perform the following TSF management functions:<br>• Configure EXE date and time;<br>• Control port IP configuration;<br>• Edit login banner;<br>• Create certificate signing request CSR, download a CSL;<br>• Reset certificates;<br>• Import certificates;<br>• Import Trusted CA certificate;<br>• Configure console menu system timeout;<br>• Verify/Install Firmware Updates<br>• View/Edit settings for sending audit data to the Syslog Server<br>• View/Edit authentication failure parameters<br>• Unlock a locked user after the login failure threshold is exceeded<br>The following can only be performed from the console interfaces:<br>• Login to local console;<br>• Change Linux password for console account "customer";<br>• Zeroize all Critical Security Parameters (CSP);<br>The TSF can also be managed from an optional MAGNUM system, which is a trusted IT entity. When a MAGNUM system is used, the server copies configuration updates to the TSF.<br>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/session timeout/generate certificate request/system reboot, etc.<br>Administrators can administer EXE remotely through its web interface, which runs on HTTPS. The web interface supports a broader set of the configuration settings that include configurations for certificate imports, syslog server, route mapping, etc.<br>The administrative interfaces provided by the TSF do not allow any of these functions to be accessed by unauthenticated or unauthorized users. |
| FPT_SKP_EXT.1 | The TSF stores cryptographic keys in a directory in flash memory. As there is no command line access, users cannot gain any direct access to these files. |
| FPT_APW_EXT.1 | The TSF does not store plaintext password. 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. |
| FPT_TST_EXT.1 | The TSF performs the following hardware self-tests at power-on: |

| TOE SFR | Rationale |
|---|---|
| | • firmware integrity check that compares the SHA256 checksum of the loaded firmware with a permanently stored hash value;<br>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:<br>• Cryptographic library tests:<br>o SHA-256 KAT<br>o HMAC-SHA-256 KAT<br>o AES 128 GCM Encrypt and Decrypt KAT<br>o RSA 4096 SHA-256 Sign and Verify KAT<br>o ECDSA Pairwise Consistency Test<br>o DRBG AES-CTR-256 KAT (invoking the instantiate, reseed, and generate functions)<br>After loading the image, a hash value is computed from the memory partition containing the image. This hash value is compared with a pre-stored hash value at another location on flash. The two hash values must match for the boot process to succeed.<br>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 is the correct image and that cryptographic functions are performing appropriately. If failures are seen by the Administrator, they should be immediately corrected. |
| FPT_TUD_EXT.1 | 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 release from Evertz. Operators may verify the current version with the web interface.<br>The current firmware version is displayed on both webpage and in serial console menu. Inactive versions can be manually set to active by the Security Administrator. Firmware updates are done from the EXE webpage interface under "upgrade".<br>During a firmware upgrade, EXE will first verify the HMAC of new firmware code with a local stored public key. The TSF does not provide an interface to change the local stored public key to administrators. When HMAC verification passes, EXE will verify the firmware binary header with an Evertz-defined proprietary format. If there is no mismatch, the new firmware code will be uploaded and parsed for a valid digital signature.<br>A verification of the firmware's digital signature is performed next. A hashed-value of the images is generated and then signed with Evertz's private key. The result file (signature) is included in the firmware package together with the actual firmware binary. During upgrade, the signature file is first decrypted using the public key stored on EXE, then the hashed value is re-calculated from the uploaded image binary file and then compared with the decrypted hash value. These hashes must match for this validation to succeed. If successful, the firmware is overwritten and the TOE is updated to the new version. If unsuccessful, the firmware update file is rejected and an error is displayed. |

| TOE SFR | Rationale |
|---|---|
| | Once the upgrade is complete, the administrator must manually change the “Next Boot Image” value from the current boot image to the newly installed image. The administrator must manually reboot for the new update to take effect. |
| FPT_STM_EXT.1 | Timestamps found in auditable log events use the system clock on EXE. Administrators can set the system time clock through serial port console menu after each card reboot.<br>Other functions which make use of timestamps include verification of X.509 certificate validity periods.<br>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. |
| FTA_SSL_EXT.1<br>FTA_SSL.3<br>FTA_SSL.4 | 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.<br>Administrators may terminate their own sessions by clicking “Logout” at the upper right hand of the web interface or typing “logout” or “exit” to exit the console. |
| FTA_TAB.1 | The TSF enables Security Administrators to alter the warning banner by navigating to the Main men, then Perpetual User License Agreement tab on the web browser. 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.<br>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. |
| FTP_ITC.1 | The TSF communicates with the external syslog server using Syslog over TLSv1.2 as described in the descriptions of FAU_STG_EXT.1 and FCS_TLS* above. The TSF initiates the trusted channel with the Syslog.<br>The TSF communicates with a MAGNUM 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. |
| 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. |

**Table 16 TOE Summary Specification SFR Description**

# 7 Terms and Definitions

| Abbreviations/Acronyms | Description |
|---|---|
| AES | Advanced Encryption Standard |
| ASI | Asynchronous Serial Interface |
| CBC | Cipher Block Chain |
| CC | Common Criteria |
| CCRA | Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security |
| CO | Cryptography Officer |
| CTR | Counter (mode) |
| DHE | Diffie-Hellman Exchange |
| DNS | Domain Name Service |
| DOD | Department of Defense |
| DRBG | Deterministic Random Bit Generator |
| ECDHE | Elliptic Curve Diffie-Hellman Exchange |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| Gb | Gigabit |
| GCM | Galois/Counter Mode |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| IP | Internet Protocol |
| KVM | Keyboard, Video, Mouse |
| NIAP | National Information Assurance Partnership |
| NDPP | Network Device Protection Profile |
| OE | Operational Environment |
| OSP | Organizational Security Policy |
| PP | Protection Profile |
| RFC | Request For Comment |
| RJ-45 | Radio Jack (45) |
| RS-232 | Recommended Standard 232 |
| RSA | Rivest-Shamir-Adelman |
| RU | Rack Unit |
| SDI | Serial Digital Interface |
| SDTI | Serial Digital Transport Interface |
| SAR | Security Assurance Requirement |
| SFP | Small Form-Factor Pluggable |
| SFR | Security Functional Requirements |
| SHA | Secure Hash Algorithm |
| SNMP | Simple Network Management Protocol |
| SPD | Security Policy Database |
| SSH | Secure Shell |
| ST | Security Target |
| Tb | Terabit |
| TLS | Transport Layer Security |
| TOE | Target of Evaluation |
| TSF | Target Security Function |
| TSFI | TSF Interface |
| TSS | TOE Summary Specification |
| USB | Universal Serial Bus |
| VGA | Video Graphics Array |

**Table 17 TOE Abbreviations and Acronyms**