MAGNUM-HW-C-CC

Security Target

Acumen Security, LLC.

Document Version: 2.2

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
1.0	August 9, 2019	Initial Draft
2.0	October 10, 2019	Updated from evaluator and validator comments as well as adding new TDs
2.1	December 13, 2019	Updated based on validator comments
2.2	December 20, 2019	Updated based on 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	MAGNUM-HW-C-CC Security Target
ST Version	2.2
ST Date	December 20, 2019
ST Author	Acumen Security, LLC.
TOE Identifier	MAGNUM-HW-C-CC
TOE Hardware	MAGNUM- HW-C-CC
TOE Firmware Version	MAGNUM-SDVN v19.10.1
TOE Developer	Evertz Microsystems Ltd. 5292 John Lucas Drive Burlington, Ontario 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 device is the Evertz MAGNUM-HW-C-CC which includes the MAGNUM-HW-C-CC (1 RU) running MAGNUM-SDVN firmware v19.10.1. The MAGNUM-HW-C-CC serves as the primary user and network interface device for the MAGNUM control application.

1.2.2 TOE Usage

Evertz MAGNUM software is a custom-developed application written primarily in python. MAGNUM operates as a combination of an application layer and as part of the integrated Linux platform stack, using a customized Ubuntu operating system. The TOE version of MAGNUM is only operable on Evertz-provided platforms and hardware.

MAGNUM serves as the control interface for Evertz's proprietary IPX media streaming switch fabric that allows the general user to establish, change, and tear down multicast IP video streams. MAGNUM may also serve as a general control interface for similar Evertz and third-party systems and devices.

Equipment to prepare video for IP transport, or to convert it into other video formats, 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 is an infrastructure network device that provides secure remote management, auditing, and updating capabilities. The TOE provides secure remote management using an HTTPS/TLS web interface and an SSH command line interface. The TOE generates audit logs and transmits the audit logs to a remote syslog server over a mutually authenticated TLS channel. The TOE verifies the authenticity of software updates by verifying the digital signature prior to installing any update.

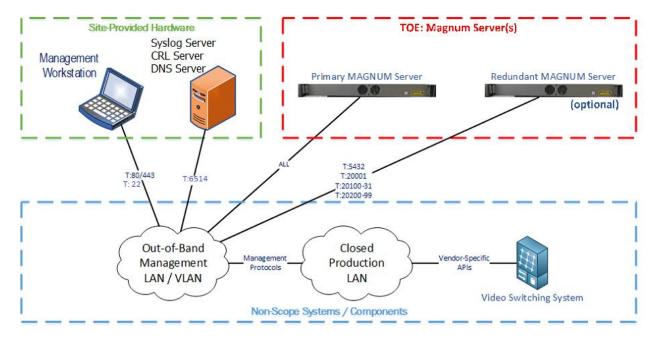The scope of the evaluated functionality includes the following,
- Secure remote administration of the TOE via TLS and SSH

- Secure Local administration of the TOE
- Secure connectivity with remote audit servers
- Secure access to the management functionality of the TOE
- Identification and authentication of the administrator of the TOE

No other functionality is included within the scope of this evaluation.

MAGNUM issues commands (via dedicated internal API) to Evertz's proprietary IPX switching fabric and other production endpoints for the purpose of initiating, maintaining, and tearing down virtual routing paths. The MAGNUM-HW-C-CC device serves as the primary operational and administrative management interface to the closed multicast switching environment.

Users and administrators may access MAGNUM software via direct connection using a terminal session. Administrators only may access MAGNUM via a dedicated management workstation operating over an Out-of-Band Management (OOBM) network. Sites may close this OOBM network or may operate MAGNUM within an existing OOBM, as long as the topology is compliant with the security parameters listed below.

Figure 1 TOE Topology

1.3   TOE IT Environment

MAGNUM is a device/component management application specifically designed to operate broadcast and enterprise video switching/routing equipment. Typically, such video equipment is deployed either in an extended series of point-to-point, non-Ethernet connections (such as optical fiber, Serial Digital Interface (SDI), High Density Multimedia Interface (HDMI), HDBaseT, or similar video-specific interfaces) or via an integrated, Ethernet-based switch fabric such as Evertz's IPX product.

6

MAGNUM does not have the ability to route multimedia stream data and operates exclusively via OOBM. MAGNUM issues operational commands to production video equipment and serves as the primary interface for production devices to report to external services, such as logging tools.

MAGNUM may lie within the same production network as the video equipment or it may be on an isolated network. In the case of non-Ethernet connectivity for the video system, MAGNUM may be located on a standard production network at the discretion of the site/mission owner.

It is not necessary for MAGNUM to be located on an enterprise OOBM network, although sites may choose to do so.

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) - Supporting Syslog over TLS (RFC 5425) - Acting as a TLSv1.2 server - Supporting Client Certificate authentication - Supporting at least one of the following 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
Management workstation with web browser	Yes	- Supported browser: Chrome or Safari - Supporting TLSv1.2 - Supporting Client Certificate authentication - Supporting at least one of the following ciphersuites: - 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
CRL Server	Yes	Conformant with RFC 5280
DNS Sever	Yes	Conformant with RFC 1035

Table 2 IT Environment Components

1.4 TOE Architecture

The TOE consists of the MAGNUM-HW-C-CC hardware, running MAGNUM-SDVN firmware version 19.10.1. Evertz MAGNUM software is a Linux-based application that can be provided on many computing platforms, including Evertz-manufactured control panels. This TOE requires that MAGNUM software be deployed on a dedicated, custom-built Evertz-owned platform (model MAGNUM-HW-C-CC); collectively, this product is known as MAGNUM-HW-C-CC. The MAGNUM-HW-C-CC features an embedded Ubuntu Operating System (OS) v18.04.2 with kernel 4.15.0 on an Intel® Xeon® D-1541 processor and includes the following interfaces:

- 1 x 1 Gigabit Ethernet Eternal Traffic Port
- 2 x RJ45 ports
- 1 x RJ45 dedicated IPMI LAN port
- 2 x USB ports (USB 3.0 connectors)
- VGA Output
- Dual power supply

For local console access a standard video display would be connected via the VGA output and a keyboard and mouse would be connected via USB.

The MAGNUM-HW-C-CC device has three components: the MAGNUM software package, an embedded customized Ubuntu operating system, and an Evertz customized hardware platform.

The guidance documentation that is part of the TOE is listed in Section 1.4.3.

1.4.1   Logical Scope of the TOE
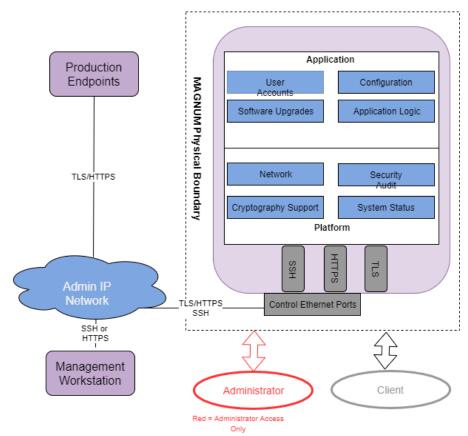The figure below depicts the logical scope of the TOE.

Figure 2 TOE Logical Scope and Workflow

The TOE supports the following functionality:

- Security Audit
- Cryptographic Support

8

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

1.4.2 Security Functions provided by the TOE

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

1.4.2.1 Security Audit

The TOE generates audit records for security relevant events. Audit data are stored internally and are only accessible to privileged administrators. The TOE supports role-based access control (RBAC) for authentication and authorization to management and security functions.

The TOE also supports sending audit records to a remote Syslog server. Audit records sent to the remote server are protected by a TLS connection. Each audit record includes identity (username, IP address, or process), date and time of the event, type of event, and the outcome of the event.

1.4.2.2 Cryptographic Support

The TOE includes an OpenSSL library 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, and SSH 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.

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

Table 3 TOE Cryptographic Protocols

Each of these cryptographic algorithms have been validated for conformance to the requirements specified in their respective standards, as identified below.

Algorithm	Standard	CAVP Certificate #	Processors
AES 128/256-bit CBC, CTR, GCM	IOS 19772 (GCM) IOS 10116(CTR)	C1168	Intel® Xeon® D-1541
CTR DRBG using AES 256	ISO/IEC 18031:2011	C1168	Intel® Xeon® D-1541
EC-DH	NIST SP 800-56A (key establishment)	C1168	Intel® Xeon® D-1541
ECDSA with NIST curves P-256, P384	FIPS PUB 186-4, "Digital Signature Standard (DSS), Appendix B.4	C1168	Intel® Xeon® D-1541
HMAC-SHA-1/256/384/512	ISO/IEC 9797-2:2011	C1168	Intel® Xeon® D-1541
SHA-1/256/384/512	ISO/IEC 10118-3:2004	C1168	Intel® Xeon® D-1541
RSA 2048-, 3072-, 4096-bit	FIPS PUB 186-4 (key generation)	C1168[1]	Intel® Xeon® D-1541
RSA 2048-, 3072-, 4096-bit	ISO/IEC 9796-2 (digital signature generation and verification)	C1168	Intel® Xeon® D-1541

Table 4 CAVP Algorithm Testing References

1.4.2.3   Identification and Authentication

The TOE authenticates administrative users using a username/password combination. The TOE does not allow access to any administrative functions prior to successful authentication. The TOE validates and authenticates X.509 certificates for all certificate uses.

The TOE supports passwords consisting of alphanumeric and special characters and enforces minimum password lengths. The TSF supports certificates using RSA signature algorithms. Certificates are used to authenticate trusted channels, not administrators. The TOE only allows users to view the login warning banner prior to authentication. Remote administrators are locked out after a configurable number of unsuccessful authentication attempts.

1.4.2.4   Security Management

The TOE allows users with the Security Administrator role to administer the TOE over a remote web UI, remote CLI, or a local CLI. These interfaces do not allow the Security Administrator to execute arbitrary commands or executables on the TOE. Security Administrators can manage connections to an external Syslog server, as well as determine the size of local audit storage.

1.4.2.5   Protection of the TSF

The TOE implements several self-protection mechanisms. This protection includes self-tests to ensure the correct operations of cryptographic functions. Firmware upgrades, performed by a Security Administrator, must pass two authentication tests. The TOE does not provide an interface for the reading of secret or private keys. The TOE ensures timestamps, timeouts, and certificate checks are accurate by maintaining a real-time clock.

---

[1] RSA key generation 4096-bit is vendor affirmed

1.4.2.6   TOE Access

The TOE can be configured to display a warning and consent banner when an administrator attempts to establish an interactive session over the CLI (local or remote) or remote web UI. The TOE also enforces a configurable inactivity timeout for remote administrative sessions.

1.4.2.7   Trusted Path/Channels

The TOE uses TLS to provide a trusted communication channel between itself and remote. The trusted channels utilize X.509 certificates to perform mutual authentication. The TOE initiates the TLS trusted channel with the remote server.

The TOE uses HTTPS/TLS and SSH 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.

1.4.2.8   Unevaluated Functionality

The nature of the physical network connection is considered outside the scope of the TOE, as the available network elements (IP switches, IP routers, etc.) which may be used in establishing that link are site-specific. Evertz stipulates that any connection must meet organizationally specific security requirements for the location(s) where the equipment is deployed.

The purpose of the MAGNUM control system is to control a variety of video equipment, including:
- Analog Video Routing Switches
- Digital Video Routing Switches (SDI)
- Digital Video Routing Switches (ASI)
- Digital Video Routing Switches (IP)
- Video Tally Switches
- KVM Routing Switches
- Audio Routing Switches
- Video Master Control Systems
- Video Branding Systems
- Multiviewers
- Video Transport Systems

These will be referred to as "video switches." These are outside the scope of the TOE.

1.4.2.9   Excluded Functionality

The TOE includes the following functionality that may not be enabled or used in in the CC evaluated configuration:
- External Authentication Servers for administrator authentication
- SNMP traps

11

1.4.3 TOE Documentation

The table below lists the TOE guidance documentation. CC1 and CC2 are delivered via hardcopy to customers with delivery of the TOE. CC3 and ST are provided in .pdf form on the NIAP portal.

Reference	Title	Version	Date
[CC1]	MAGNUM User Manual	1.3	August 2014
[CC2]	MAGNUM-SDVN Management and Control of Evertz IP Switch Fabrics and Gateways User Manual	0.1	November 2014
[CC3]	MAGNUM-SDVN Security Administration Manual	26	December 20, 2019
[ST]	Evertz MAGNUM-HW-C-CC Security Target	2.2	December 20, 2019

Table 5 TOE Guidance Documents

1.4.4 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 2, 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)
0453 – NIT Technical Decision for Clarify authentication methods SSH clients can use to authenticate SSH	No	FCS_SSHC_EXT is not claimed.
0452 – NIT Technical Decision for FCS_(d)TLSC_EXT.X.2 IP addresses in reference identifiers	Yes	
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	
0449 – NIT Technical Decision for Identification of usage of cryptographic schemes	Yes	
0448 – NIT Technical Decision for Documenting Diffie-Hellman 14 groups	No	Diffie-Hellman group 14 is not selected.
0447 – NIT Technical Decision for Using 'diffie'hellman-group-exchange-sha256' in FCS_SSHC/S_EXT.1.7	Yes	
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	Yes	
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	Yes	
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.

Identifier	Applicable	Exclusion Rationale (if applicable)
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	Yes	
0408 – NIT Technical Decision for local vs. remote administrator accounts	Yes	
0407 – NIT Technical Decision for handling Certification of Cloud Deployments	Yes	
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	Yes	
0397 – NIT Technical Decision for Fixing AES-CTR Mode Tests	Yes	
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 6 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 7 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. 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 8 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 9 OSPs

17

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	TOE Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. 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 (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 10 Objectives for the Operational Environment

18

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 4, dated: September 2012 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_SSHS_EXT.1	SSH Server Protocol
FCS_TLSC_EXT.2	TLS Client Protocol with authentication
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 11 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 12*.

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 12*.

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_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.2	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 Any addition, replacement or removal of trust anchors in the TOE's trust store.	Reason for failure of certificate validation 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.

21

Requirement	Auditable Events	Additional Audit Record Contents
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
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)	The termination of a local session by the session locking mechanism.	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. Termination of the trusted channel. 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. Termination of the trusted path. Failure of the trusted path functions.	None.

Table 12 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 [*drop new audit data*] 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" [P256, P384] 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 Specification Version 2.1";
- *Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation 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 [a pseudorandom pattern using the TSF's RBG, zeros]*;*

23

        o   instructs a part of the TSF to destroy the abstraction that represents the key]

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, CTR, 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, CTR 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, 4098 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, SHA-512] ~~and cryptographic key sizes [~~*~~assignment: cryptographic key sizes~~*~~]~~ and **message digest sizes [160, 256, 384, 512] 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-SHA1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes [*160 bits, 256 bits, 512 bits used in HMAC*] **and message digest sizes [160, 256, 384, 512] 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 source] 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_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1

The TSF shall implement the SSH protocol that complies with RFC(s) [4251, 4252, 4254, 5656, 6187, 6668].

FCS_SSHS_EXT.1.2

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [password-based].

FCS_SSHS_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [*263K*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4

The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [aes128-ctr, aes256-ctr].

FCS_SSHS_EXT.1.5

The TSF shall ensure that the SSH public-key based authentication implementation uses [*ssh-rsa, rsa-sha2-256, rsa-sha2-512*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6

The TSF shall ensure that the SSH transport implementation uses [hmac-sha2-256, hmac-sha2-512] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7

The TSF shall ensure that [ecdh-sha2-nistp256] *and* [ecdh-sha2-nistp521, ecdh-sha2-nistp384] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8

The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

ST Application Note
FCS_SSHS_EXT.1.5 was updated based on TD0424.

5.2.2.11   FCS_TLSC_EXT.2 TLS Client Protocol with authentication

FCS_TLSC_EXT.2.1

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.2.2

The TSF shall verify that the presented identifiers of the following types: [identifiers defined in RFC 6125, IPv4 address in SAN] are matched to reference identifiers.

ST Application Note
FCS_TLSC_EXT.2.2 was updated based on TD0452.

FCS_TLSC_EXT.2.3

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.2.4

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

FCS_TLSC_EXT.2.5

The TSF shall support mutual authentication using X.509v3 certificates.

5.2.2.12   FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1

26

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 [4096 bits], generate EC Diffie-Hellman parameters over NIST curves [secp256r1], 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 10*] 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 [*unlocks the* user] is taken by an Administrator; prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].

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: ["!", "@", "#", "$", "%", "^", "&", "*", "(", ")",["~", "", "", "+", "-", ",", ".", "/", "\", ":", ";", "<", "=", ">", "?", "[", "]", "_", "`", "{", "|", "}", [space]]];
  b) Minimum password length shall be configurable to between [*8*] and [*15*] characters.

27

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, SSH public key-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 certificate 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 [*not 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 [determine the behavior of, 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 [<u>digital signature</u>] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- *[*
  - *<u>Ability to configure audit behaviour</u>;*
  - *<u>Ability to manage cryptographic keys</u>;*
  - *<u>Ability to set the time which is used for time</u>-stamps;*
  - *<u>Ability to import X.509v3 certificates to the TOE's trust store</u>*

  ].

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 passwords in non-plaintext form.

FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext passwords.

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: [

- *Kernel integrity check that compares the SHA512 checksum of the kernel against permanently stored hash values*
- *Firmware integrity check that compares the SHA512 checksum of every executable and library file against permanently stored hash values*
- *Security mode verification that compares the SHA512 checksum of security-policy configuration files against stored hash values*
- *Correct operation of cryptographic functions by explicitly invoking OpenSSL's FIPS self-test*
  - *SHA-256/284/521 KAT*
  - *HMAC-SHA-256/521 KAT*
  - *AES 128 GCM Encrypt and Decrypt KAT*
  - *AES 256 GCM Encrypt and Decrypt KAT*
  - *AES 128 CTR Encrypt and Decrypt KAT*
  - *RSA 4096 SHA-256 Sign and Verify KAT*
  - *DRBG AES-CTR-256 KAT (invoking the instantiate, reseed, and generate functions)*
- *Presence of certificate and public key files*
].

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 [*no other TOE firmware/software version*].

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 switches_]** 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 [*audit 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 [*SSH, TLS, HTTPS*] to** provide a communication path between itself and **authorized** <u>remote</u> **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 <u>remote</u> **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 13 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 will provide the TOE for testing.
AVA_VAN.1	Evertz will provide the TOE for testing. Evertz will provide a document identifying the list of software and hardware components.

Table 14 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 FAU_GEN.2	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. Audit records are stored in plaintext in /var/log for each application. Each entry contains a timestamp of when the event had occurred as well as a message body with description of the event. Log entries are sorted based on chronological order. The TSF generates audit records for the following events: • Startup and shutdown of the audit function • Administrative login and logout events • Changes to TSF data related to configuration changes • Generation of a CSR and associated keypair • Installation of a certificate • Resetting passwords • Failure to establish a HTTPS/TLS session • Failure to establish a TLS session • All use of the identification and authentication mechanism (local and remote connections to the TSF) • Unsuccessful attempts to validate a certificate • Initiation of a software update • Result of a software update • Changes to the time • Modification of the behavior of the TSF • Failure of self-tests • Initiation and termination of the trusted channel • Initiation and termination of the trusted path • Attempts to unlock an interactive session • Termination of a session by the session locking mechanism • Starting and stopping services Each audit record includes the date and time, type, subject identity (IP address, hostname, and/or username), the outcome (success or failure), and any additional information specified in column three of Table 12. The TOE includes 3 different keys. When a key is destroyed or generated a log message is created and the keys are referred to as follows: • TLS keys – 'ssl/private/evertz-server.key' • SSH keys – 'ssh_host_rsa_key' The TOE only stores one of each type of key and therefore these names uniquely identify the keys stored on the TOE.
FAU_STG_EXT.1	The TOE is a standalone TOE. Audit data is sent to external syslog server through secured, mutually authenticated TLS v1.2 sessions. A Security Administrator must configure an external syslog server (IP address/TCP Port number) on the TOE. A trusted certificate chain that is used to sign syslog server's certificate must be also uploaded to MAGNUM. The trusted channel with the Syslog server is described in greater detail in the FCS_TLSC_EXT.2 description.

TOE SFR	Rationale
	MAGNUM stores all audit data locally on SSD in a 20 GB non-executable partition, protected by Linux permissions. Only authorized administrators can access the stored audit data. To keep the local audit disk partition from overflowing old audit records on the local SSD are transmitted to the audit server once a connection is available. In the unlikely event that the disk partition fills up before enough records can be rotated away new entries are dropped. The TSF protects audit data from unauthorized modification and deletion through the restrictive administrative interfaces. The filesystem of the TSF is not exposed to the administrative user over the HTTPs GUI or the local CLI. The administrative user must be positively identified and authenticated prior to being allowed to clear the local audit log or change audit settings.
FCS_CKM.1	The TSF supports 4096-bit RSA keys for generation of keys for TLS session signatures and ECDSA with NIST curves P-256 and P-384 to generate ECDH components for TLS key establishment.
FCS_CKM.2	The TOE acts as both sender and recipient for elliptic curve-based key establishment schemes that meet the following: - NIST Special Publication (SP) 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" – for FCS_TLSC_EXT.2 connections to the audit server and FCS_TLSS_EXT.1 connections to video switches. 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.2 connections to audit server and FCS_TLSS_EXT.1 connections to video switches. The TOE also supports key exchange for SSH using <u>ecdh-sha2-nistp256, ecdh-sha2-nistp521, or ecdh-sha2-nistp384 for FCS_SSHS_EXT.1.</u> In the case of a decryption error, the TOE response is dependent on the stage of the connection process. If the connection has not been established, the TOE prevents a connection from occurring. If the connection has already been established, the TOE drops the packet(s) in question and logs the error internally. To address the issue of side-channel attacks, the TOE does not reveal the particular error that occurred through other channels, either through message content or timing variations.
FCS_CKM.4	The TSF overwrites keys with random data followed by overwriting the contents with zeros. After each write operation, MAGNUM reads the data to confirm that it only the new data is stored (as opposed to a cached or older version of the data). If this test fails, the process is repeated until it 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. Keys are stored on a separate disk partition that uses uses Linux file permission to ensure that no user or administrator access is allowed. The TOE does not provide full shell access and file permissions cannot be changed. No user has access to this partition. Keys are

TOE SFR	Rationale
	cleared when entering secure mode during device setup, and whenever the administrator selects this operation from the console. The following key is stored in the partition: • The private RSA key matching the server certificate, which is generated by the TSF and used for HTTPS web interface, and IPX and remote audit server connection; No direct interface/access is provided to view or modify the contents of these files. The TLS Session keys are zeroized from RAM when the associated TLS session is terminated. The DRBG state is zeroized using a single overwrite of zeros when the TSF is shutdown or restarted.
FCS_COP.1/DataEncryption	The TOE provides AES encryption/decryption in CBC, CTR and GCM modes with 128- and 256-bit keys.
FCS_COP.1/SigGen	The TOE supports signature verification with RSA (2048-, 3072-, 4096-bit) with SHA-256/384 and signature generation with RSA (4096-bit) with SHA 256/384 in accordance with FIPS PUB 186-4. These signatures support TLS authentication.
FCS_COP.1/Hash	Cryptographic hashing services are performed using Evertz's cryptographic module. Hashing is used for firmware integrity checks, password verification and security mode verification. The TOE provides cryptographic hashing services for. The TOE implements hashing in byte-oriented mode. The TOE uses hashing for the following security functions: • TLS connection establishment using SHA-1/256/384 • Verifying executable file checksums SHA-512 • Linux Passwords using salted SHA-512 • Key generation using SHA-256 as specified in NIST SP 800-90 DRBG
FCS_COP.1/KeyedHash	MAGNUM uses OpenSSL software that has been patched to enforce FIPS modes using special environment variables. When these are set, the TSF will not allow ephemerally generated hashes and keys to be created that do not comply with these standards. The keyed-hash message authentication is performed internally by OpenSSL when it is used to perform message authentication. For HMAC-SHA-1: • Key length: 160 bits • Hash function used: SHA-1 • Block size: 256 bits • Output MAC (message digest size): 160 bits For HMAC-SHA-256: • Key length: 256 - 512 bits • Hash function used: SHA-256 • Block size: 512 bits • Output MAC (message digest size): 256 bits

TOE SFR	Rationale

	For HMAC-SHA-384:
	
	    • Key length: 384 bits
	    • Hash function used: SHA-384
	    • Block size: 1024
	    • Output MAC (message digest size): 384 bits
	
	For HMAC-SHA-512:
	
	    • Key length: 512 bits
	    • Hash function used: SHA-512
	    • Block size: 1024 bits
	    • Output MAC (message digest size): 512 bits
	
	HMACs are used for verification of the firmware image and encrypted password files during bootup.

FCS_HTTPS_EXT.1
FCS_TLSS_EXT.1
FCS_TLSC_EXT.2
	The TSF implements the server and client sides of the HTTPs protocol according to RFC 2818 by using a TLS session in place of a TCP connection.
	
	The TSF 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. When the TSF is configured with a server certificate with an RSA key, the TSF supports following restrictive TLS ciphersuties 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
	
	MAGNUM supports cipher suites that use ECDHE or RSA keys for key exchange and RSA for authentication. These keys are generated by the OpenSSL implementation internally with OpenSSL's RSA command line utility. When acting as a TLS server, the TOE Key Exchange message parameters are 4096-bit RSA key or ECDSA over NIST curve secp256r1.
	
	MAGNUM uses CRL (certification revocation list) to check for invalid certificates. CRL files which are signed by trusted CA certificated can be imported to MAGNUM. This CRL file will be used by MAGNUM during certificate validation process to check for revocation status of the peer certificates. By default, the TOE presents the supported Elliptic Curve Extension secp256r1 in the Client Hello.
	
	MAGNUM allows configuration of reference identifier from peer it expects to connect with before connection is made. The verification against SAN-DNS peer certificate is implemented within OpenSSL.
	
	For browser-based management, MAGNUM must respond to the request presented by the user/operator browser. Administrators do not have the ability to modify the available ciphersuites, as these are hard-coded at the application layer. The [CC2]

TOE SFR	Rationale
	describes configuration procedures for the allowed SANs. When establishing a TLS connection, the MAGNUM client establishes the following reference identifiers: • Domain Name Service (DNS) in CN or SAN-DNS • IPv4 Address in SAN-IP The SAN field is mandatory when using SAN-IP. If there is no SAN-DNS field provided, the default fallback position is the Common Name (CN). When establishing reference identifiers, wildcards are supported for DNS only. MAGNUM supports wildcard in certificates. The wildcard must be in the left-most label of the presented identifier. And the wildcard 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. Certificate pinning is not used. The TSF sends the client EC Diffie-Hellman secp256r1 NIST curve. The TSF does not provide support for elliptic curves in the ClientHello message. When the Syslog server or video switch sends the Certificate Request message, the TSF replies with a Client Certificate message. The Client Certificate message includes the certificate that the Security Administrator configured to authenticate to the Syslog server. MAGNUM functions as an HTTPS server only. HTTPS is used implementation to provide a secure interactive webpage interface for remote administrative functions, and to support secure exchange of user authentication parameters during login. The internal application is "stunnel." Certificates (MAGNUM's own certificate or a trusted CA certificate) can be uploaded onto MAGNUM prior to establishing connection with peers. These certificates are used in the TLS handshaking process and is taken care of by TLS protocol implementation. The TSF will not establish the connection if the peer certificate does not successfully authenticate the peer according to X.509 authentication. When acting as a server, the TSF listens on port 443 for HTTPs connections. The TSF uses HTML over HTTPs to present the administrative users with a secure management interface. The TSF uses TLS to provide a secure connection between the TSF and remote Security Administrators. When acting as a client, the TSF uses HTTPs to establish a trusted channel with a syslog server or video switch.
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 256 bits of entropy. The TSF gathers and pools entropy from one software-based noise source: haveged and the Linux Kernel entropy. The entropy sources are discussed in greater detail in the Entropy documentation.
FCS_SSHS_EXT.1	The TSF implements SSH as a trusted channel for remote administrative connections to the CLI. Public key or password-based authentication is allowed. ssh-rsa, rsa-sha2-256, and rsa-sha2-512 are the only public key algorithms accepted for SSH connections. All

TOE SFR	Rationale
	connections using other public key algorithms are rejected. Keys are exchanged using elliptic curve Diffie Hellman with NIST curves P-256, P-384 or P-521. If an SSH client attempts a session with public key authentication and does not provide the proper key, the TOE will reject the authentication attempt and revert to password-based authentication. During an SSH session, the TOE reads the packet payload size from the TCP header to determine packets size. As packets are reassembled, the payloads are added. Any packets larger than 263KB are rejected. SSH transport is encrypted using AES CTR with key sizes of 128- or 256-bits. Data integrity is verified using HMAC-SHA256 or HMAC-SHA512. All other MAC algorithms are rejected. The TSF will rekey the SSH if the session lasts longer than 60 minutes or if more than 1GB of data have been transferred.
FIA_AFL.1	An administrator can configure the number of unsuccessful attempts a remote administrator can make before a lock-out and can configure the length of time that the remote administrator is locked out. The attempts can range between 3 and 10, with a default of 10. The length of time can be configured between 1 and 60 minutes, with a default of 15. Additionally, a different Administrator can log in and unlock the user, prior to the timeout period if needed. The TOE maintains a counter for incorrect authentication attempts for each username. If the user enters an incorrect password the configured number of times, the username is changed to a locked state. Any attempt to authenticate from a remote interface using that username is denied and an error message is shown to the user. When the lockout time has expired or an administrator unlocks the user, the administrator is allowed to authenticate to the TOE again. 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	MAGNUM enforces that passwords must meet minimum requirements (length, mix of number of lower/upper case letters, numbers as well as special characters, no common dictionary words. etc). The special characters the TSF supports include *: "~", "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "''", "'", "+", "-", ",", ".", "/", "\", ":", ";", "<", "=", ">", "?", "[", "]", "_", "`", "{", "|", "}", [space].* Administrators can configure a minimum password length between 8 and 15 characters. Guidance documentation recommends that Administrators set the minimum password length to 15 characters.
FIA_UIA_EXT.1 FIA_UAU_EXT.2	The TSF displays a warning banner after the user enters their username, but before the password prompt will accept login credentials from a user. This applies to direct console users as well as web users. Authentication is based on username/password for the web interface and local console. Remote access of SSH can use password or SSH public key-based authentication. The TOE does not expose any interface, through any access method prior to successful login. Console user's passwords are verified through the PAM module provided by Linux. Web user's passwords are verified against a PostgreSQL database that stores (hashed) values. If a password does not match, the user is not granted access. If the passwords do match, the user is granted access according to their role. SSH user's with public key-based authentication must first have their public key uploaded to the TOE. The user can then start an SSH session with the TSF using the public key. The

TOE SFR	Rationale
	key is verified against the stored key. If the keys match, access is granted. If the keys do not match, access is not granted, and the user is presented with a username and password prompt.
FIA_UAU.7	When the user is entering their password over the local console, the TSF shows only asterisks ("*").

FIA_X509_EXT.1/Rev	MAGNUM uses CRLs to validate certificates. When the TOE acts as a TLS client, the video switch or syslog server's certificates are validated during the TLS session connection handshake. Certificates are also checked for revocation when loaded onto the TOE. When acting as a TLS server, the TOE validates certificates when presented by the client. MAGNUM first checks Certificate Authorities (CAs), then CRLs, then SANs. The TOE verifies that the certificates presented as a TLS server must contain the TLS server extended key usage, TLS client certificates must have the TLS client extended key usage. The TOE does not support certificates for trusted updates or OCSP. This validation includes revocation checking for the full certificate chain regardless of whether the full chain or only a leaf certificate is presented.
	
	MAGNUM only supports certificates that have been loaded by an authorized system administrator within the local Evertz network environment. As a purpose-built ecosystem, MAGNUM will not operate non-Evertz hardware. Administrators should ensure that the CRL reflects the certificates loaded onto the IPX or other Evertz hardware which the system is intended to manage.
	
	For an expired certificate, MAGNUM will deny the connection. MAGNUM also uses a CRL to verify whether the certificate or intermediate CA certificate has been revoked. During session establishment with MAGNUM, any byte modification in the certificate will lead to the failure of connection.
	
	The TSF additionally verifies:
	• Each certificate (other than the first certificate) in the certificate chain has the Subject Type=CA flag set.
	• Each certificate is signed by:
	  o a certificate in the certificate chain, or
	  o a trusted root CA that has been installed in the TSF

FIA_X509_EXT.2	Instructions for configuring MAGNUM to operate with X.509 certificates are found in the [CC2] document.
	
	MAGNUM only has one certificate to present. If the certificate fails a validity check, the connection attempt fails and the trusted channel is not established.
	
	The following trusted channels are supported:
	
	• MAGNUM-HW-C-CC to another MAGNUM-HW-C-CC
	• MAGNUM-HW-C-CC to a remote syslog server
	• MAGNUM-HW-C-CC to an Evertz video switch compatible with Common Criteria. As of this writing these include:
	  o MMA10G-IPX-16
	  o MMA10G-IPX-32
	  o MMA10G-IPX-64
	  o MMA10G-EXE16
	  o MMA10G-EXE26
	  o MMA10G-EXE36

TOE SFR	Rationale
	o MMA10G-IPX-128 MAGNUM may also control 3rd-party devices as long as such devices support TLS v1.2. In such cases, MAGNUM can support a trusted channel to such devices. The configuration and deployment of 3rd-party devices lies outside the scope of the TOE and this ST. If the TOE is unable to reach the CRL DP it will not accept the certificate and the session associated with the certificate will be denied.
FIA_X509_EXT.3	MAGNUM uses its OpenSSL based cryptographic module to generate a Certificate Request Message. This requires the specification of the public key, Common Name, Organization, Organizational Unit, and Country. This information is configurable via the console admin interface. MAGNUM uses the following key usage and extended key usage parameters: • keyUsage = critical,nonRepudiation,digitalSignature,keyEncipherment • ExtendedKeyUsage = clientAuth,serverAuth MAGNUM uses its OpenSSL based cryptographic module to verify certificates when the TOE is configured in a security mode to verify certificates by a Certificate Authority (CA). MAGNUM requires all certificates in the chain to be presented by the peer during connection attempts.

FMT_MOF.1/Functions FMT_MOF.1/ManualUpdate FMT_MTD.1/CoreData FMT_MTD.1/CryptoKeys FMT_SMF.1 FMT_SMR.2	The TSF 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. The TSF 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 TSF implements the Security Administrator role to authorized administrators of the TOE. The TSF allows the Security Administrators to administer the TSF via the CLI (local and remote) and a web UI. The TSF permissions restrict access to these management functions to users that have been identified, authenticated, and authorized with the Security Administrator role. The web UI and CLI allow the Security Administrator to perform the following TSF management functions:
	 • System Setup
	   o Entering High Security Mode
	   o Disabling USB Storage on the BIOS
	   o Changing Connection in Security Mode
	   o Bypass Options (Not Recommended)
	   o Full Data Purge
	 • Administrator Log In/Out of:
	   o Local Terminal
	   o Remote Terminal
	 • Web Interface Configuration:
	   o Date & Time
	   o IP Addresses
	   o Clustering
	   o Remote Audit Servers
	   o Session Timeout
	 • Transferring Files
	   o Using FTPS
	   o Using SFTP

TOE SFR	Rationale

	o Using SCP
	  • Editing the Login Banner
	  • Keys
	      o Importing a Public Key
	      o TLS Key Reset
	  • Cluster Key Import / Export / Reset
	     Certificates
	      o Create Certificate Signing Request
	      o Import / Export / Show Server Certificates
	      o Import / Export / Show Trusted Certificates
	      o Import / Show / Remove Certificate Revocation List
	  • Allowed Subject Alt Names
	      o DNS
	      o IP
	      o E-Mail
	  • Administer Passwords
	      o Set Password Minimum Length
	      o Linux User
	      o Web User
	          ▪ Add / Delete
	          ▪ Change Web Users' Passwords
	  • Audits
	      o Export Logs
	  • Firmware
	      o Check Firmware Version
	          ▪ From Terminal
	          ▪ From Web
	  • Upgrade
	
	The TOE is configured with specific user groups that can perform specific tasks. Only those in the admin group are able to access and perform updates. The filesystem ownership under Linux only allows certain users and groups to access the filesystem. Only authorized administrators can access the TOE's trust store and modify or delete certificates within the trust store. So, non-privileged users are not able to update the system files. Command line access is restricted such that regular users do not have access to command line scripts used to manage MAGNUM.
	
	The web admin and console admin user are statically created on the system. These users cannot be removed from the system.
	
	Administrator roles are statically assigned. The users admin, etservice, etdev, and the web admin are all in the Administrator role. Users created by the web interface (i.e. web users) are implicitly, automatically assigned into the ("regular") User role.
	
	Administrators can use console admin interface to administer the system locally or remotely using SSH. The web administrator can also administer MAGNUM over HTTPS.

FPT_SKP_EXT.1	MAGNUM uses Linux file permissions to only allow the appropriate services programmatic access to protect private keys from being read. The TOE's keys associated with its certificate for TLS are stored in a disk partition with file permissions that do not allow user or administrator access. None of these services have methods to expose the key beyond their immediate use.

TOE SFR	Rationale
FPT_APW_EXT.1	Passwords are the authentication data stored by the TOE. The TSF does not store plaintext password. The salted SHA-512 hash of the password is saved to disk (using the Linux PAM cracklib module). Passwords for users of the web interface are stored in a PostgreSQL database and obfuscated using a salted Blowfish hash. Both the password file and the database reside on the filesystem, which is access controlled through Linux file permissions. MAGNUM also uses Linux permissions to prevent accessing the obscured forms of the passwords.

FPT_TST_EXT.1	The firmware is validated in the following three ways on startup:
	
	• The bootloader verifies a SHA-512 checksum of the kernel image before loading the image
	• MAGNUM invokes OpenSSL to display its version, which will trigger the built-in self-tests. This ensures that the crypto module has not been tampered with. These self-tests include:
	    ○ SHA-256/284/521 KAT
	    ○ HMAC-SHA-256/521 KAT
	    ○ AES 128 GCM Encrypt and Decrypt KAT
	    ○ AES 256 GCM Encrypt and Decrypt KAT
	    ○ AES 128 CTR Encrypt and Decrypt KAT
	    ○ RSA 4096 SHA-256 Sign and Verify KAT
	    ○ DRBG AES-CTR-256 KAT (invoking the instantiate, reseed, and generate functions)
	• MAGNUM verifies SHA-512 chesksums of all non-configuration files, including executable and shared object files.
	
	These tests verify that TOE firmware has not been modified and all cryptographic functions are working correctly.

FPT_TUD_EXT.1	The MAGNUM server is typically deployed in a closed network without direct access to the internet. In these instances, Administrators are required to contact Evertz to receive notification of production updates directly or via email blast. Operators may verify the current version using the CLI menu 'Version' or on the web interface Config Management->Current System Info. Customers requiring secure delivery for site policy can request secure courier delivery of software updates. Digital delivery may be provided via File Transfer Protocol Secure (FTPS) using signed and hashed code. Instructions for FTPS transfer are found in [CC2] in the Transferring Files in High Security Mode section. When the administrator selects the update file the TSF will ask if the file should be installed. When the administrator selects [yes] the TSF automatically verifies the digital signature prior to installing the update. In the event that an update file fails verification the update is rejected and an appropriate audit record is generated.
FPT_STM_EXT.1	MAGNUM provides accurate timestamps that can be updated via manual configuration by the administrator. System time is used to provide accurate time/date stamps on audit records, to track administrator inactivity and for the validation of X.509 certificates used in TLS communications.
FTA_SSL_EXT.1 FTA_SSL.3 FTA_SSL.4	MAGNUM has a configurable timeout that can be modified using the console admin interface. The timeout is 15 minutes in secure mode, adjustable to anywhere between 1

TOE SFR	Rationale
	and 60 minutes. When a timeout occurs, the user's session is terminated and the user is logged out of the system. This applies to both console and web interactive sessions. On a terminal, select "Logout" from the console admin interface.
FTA_TAB.1	MAGNUM is managed by the local or remote console admin interface or through the HTTPS web interface. From the console admin interface, under the Security menu, select "Edit Login Banner". When modifications are complete, press Ctrl+X to save and exit the editor. The TSF presents the access banner prior to authentication when a user connects to the remote web UI or local console CLI described in the FIA_UIA_EXT.1, FIA_UAU_EXT.2 description. Administrators access the console through directly connected USB keyboard and VGA monitor.
FTP_ITC.1	A trusted channel using TLS is created when certificates are imported into MAGNUM and the other device. Thus, when using the stunnel program to communicate with other services over TLS, the trusted certificate verifies the validity of the communication via mutual X.509 certificates. Trusted channels are established between the TOE and a remote audit server and video switches. The TOE initiates the connection for remote audit servers and video switches.
FTP_TRP.1/Admin	MAGNUM only communicates with Administrative Users via Trusted Paths. For remote administration this is restricted to a GUI over HTTPS or the command line over SSH. MAGNUM uses encryption and restricts the choices of ciphers, hashes, and key-exchange algorithms to those allowed by the NDcPP.

Table 15 TOE Summary Specification SFR Description

7   Terms and Definitions

Abbreviations/Acronyms	Description
10G	10 Gigabit
AES	Advanced Encryption Standard
API	Application Programming Interface
ASLR	Address Spece Layout Randomization
AV	Audio-Video, Audiovisual
CA	Certificate Authority
CBC	Cipher Block Chain
CC	Common Criteria
CMC	CBC-mask-CBC
CO	Cryptography Officer
CRL	Certificate Revocation List
CTR	Counter (mode)
CWDM	Coarse Wave Division Multiplexing
DEP	Data Executable Prevention
DFB	Distributed Feedback
DHE	Diffie-Hellman Exchange
DN	Distinguished Name
DNS	Domain Name Service
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
DVI	Digital Video Interface
DWDM	Dense Wave Division Multiplexing
ECDHE	Elliptic Curve Diffie-Hellman Exchange
ECDSA	Elliptic Curve Digital Signature Algorithm
EMX	Evertz Modular Crosspoint
FIPS	Federal Information Processing Systems
FTPS	File Transfer Protocol Secure
GB	Gigabyte
GCM	Galois/Counter Mode
HDMI	High Density Multimedia Interface
HID	Human Interface Device
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I/O	Input / Output
IAW	In Accordance With
IP	Internet Protocol
IPTV	IP Television
IPX	Internet Protocol Crosspoint
IT	Information Technology
km	Kilometer(s)
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
max	Maximum
NDcPP	Network Device collaborative Protection Profile
NIST	National Institute of Standards and Technology
NLE	Non Linear Editor, Non Linear Editing
nm	Nanometer(s)

Abbreviations/Acronyms	Description
NMS	Network Management System
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OE	Operational Environment
OID	Object Identifier
OOBM	Out of Band Management
OS	Operating System
PP	Protection Profile
PPAS	Protection Profile for Application Security
PPS	Ports. Protocols, and Services
RA	Registration Authority
RBAC	Role Based Access Control
RFC	Request For Comment
RJ-45	Radio Jack (45)
RS-232	Recommended Standard 232
RSA	Rivest-Shamir-Adelman
RU	Rack Unit (1.75")
SAN	Subject Alternative Name
SDI	Serial Digital Interface
SDVN	Software Defined Video Networking
SFP	Small Form-Factor Pluggable
SHA	Secure Hash Algorithm
SMF	Single Mode Fiber
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
T:	TCP / Transmission Control Protocol
TLS	Transport Layer Security
U:	UDP / User Datagram Protocol
USB	Universal Serial Bus
VGA	Video Graphics Array
VLAN	Virtual Local Area Network
WRT	With Respect To

Table 16 TOE Abbreviations and Acronyms

Abbreviations/Acronyms	Description
CC	Common Criteria
CCRA	Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security
DOD	Department of Defense
NIAP	National Information Assurance Partnership
OSP	Organizational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFP	Security Function Policy
SPD	Security Policy Database
ST	Security Target
TOE	Target of Evaluation

Abbreviations/Acronyms	Description
TRRT	Technical Rapid Response Team
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification

Table 17 CC Abbreviations and Acronyms